



**AlphaServer ES40 and
AlphaStation ES40
User Interface Guide**

Order Number: EK-ES240-UI. B01

This manual is for managers and operators of ES40 systems.

First Printing, April 1999

Revised February 2000

The information in this publication is subject to change without notice.

COMPAQ COMPUTER CORPORATION SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN, NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL.

This publication contains information protected by copyright. No part of this publication may be photocopied or reproduced in any form without prior written consent from Compaq Computer Corporation.

The software described in this guide is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

© 2000 Compaq Computer Corporation.

All rights reserved. Printed in the U.S.A.

Alpha, AlphaServer, and OpenVMS are registered in the U.S Patent and Trademark Office. COMPAQ, the Compaq logo, and Tru64 are copyrighted and are trademarks of Compaq. Linux is a registered trademark of Linus Torvalds. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Ltd. Other product names mentioned herein may be the trademarks of their respective companies.

Shielded Cables: If shielded cables have been supplied or specified, they must be used on the system in order to maintain international regulatory compliance.

Warning! This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Achtung! Dieses ist ein Gerät der Funkstörgrenzwertklasse A. In Wohnbereichen können bei Betrieb dieses Gerätes Rundfunkstörungen auftreten, in welchen Fällen der Benutzer für entsprechende Gegenmaßnahmen verantwortlich ist.

Attention! Ceci est un produit de Classe A. Dans un environnement domestique, ce produit risque de créer des interférences radioélectriques, il appartiendra alors à l'utilisateur de prendre les mesures spécifiques appropriées.

警告使用者:

這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策。

に基づくクラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

FCC Notice: The equipment described in this manual generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules, which are designed to provide reasonable protection against such radio frequency interference. Operation of this equipment in a residential area may cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference. Any modifications to this device—unless expressly approved by the manufacturer—can void the user's authority to operate this equipment under part 15 of the FCC rules.

Contents

Preface	xi
---------------	----

Chapter 1 Console Program Overview

1.1 Console Overview	1-2
1.2 Console Terminal	1-3
1.3 Console Mode	1-4
1.3.1 SRM Console.....	1-4
1.3.2 AlphaBIOS Console	1-4
1.3.3 Remote Management Console.....	1-5

Chapter 2 SRM Console

2.1 Invoking the SRM Console.....	2-2
2.2 SRM Command Overview	2-4
2.3 Management Tasks Performed from SRM.....	2-9
2.4 Getting Help on SRM Commands	2-10
2.5 Displaying the Logical Configuration	2-12
2.6 Displaying the Bootable Devices.....	2-17
2.7 Displaying the System FRUs.....	2-18
2.8 Displaying FRUs with Errors	2-21
2.9 Displaying the Memory Configuration	2-22
2.10 Displaying the PAL Version.....	2-23
2.11 Displaying the Power Status.....	2-24
2.12 Displaying the SRM Console Version	2-25
2.13 Booting an Operating System	2-26
2.14 Configuring a PCI NVRAM Module.....	2-28
2.15 Configuring RAID Devices.....	2-29
2.16 Testing the System.....	2-30
2.17 Forcing a System Crash Dump	2-32
2.18 Resuming Program Execution	2-34
2.19 Reading a File	2-35
2.20 Initializing the System.....	2-36

2.21	Initializing the Hardware Resources in an OpenVMS Galaxy	2-38
2.22	Loading the AlphaBIOS Console	2-39
2.23	Creating a Power-Up Script.....	2-40
2.24	Entering the RMC from the Local VGA Monitor	2-42
2.25	Setting and Viewing Environment Variables	2-44
2.25.1	auto_action.....	2-49
2.25.2	bootdef_dev	2-51
2.25.3	boot_file.....	2-52
2.25.4	boot_osflags.....	2-53
2.25.5	com*_baud	2-57
2.25.6	com*_flow.....	2-58
2.25.7	com1_mode.....	2-59
2.25.8	com*_modem.....	2-61
2.25.9	console.....	2-62
2.25.10	console_memory_allocation.....	2-63
2.25.11	cpu_enabled	2-65
2.25.12	ei*0_inet_init or ew*0_inet_init.....	2-67
2.25.13	ei*0_mode or ew*0_mode.....	2-68
2.25.14	ei*0_protocols or ew*0_protocols.....	2-69
2.25.15	kbd_hardware_type	2-70
2.25.16	language.....	2-71
2.25.17	lp_count.....	2-72
2.25.18	lp_cpu_mask <i>n</i>	2-73
2.25.19	lp_io_mask <i>n</i>	2-74
2.25.20	lp_mem_size <i>n</i>	2-75
2.25.21	lp_shared_mem_size	2-76
2.25.22	memory_test.....	2-77
2.25.23	ocp_text.....	2-78
2.25.24	os_type	2-79
2.25.25	pci_parity	2-80
2.25.26	pk*0_fast.....	2-81
2.25.27	pk*0_host_id	2-82
2.25.28	pk*0_soft_term.....	2-83
2.25.29	tt_allow_login.....	2-85
2.26	Ensuring Console Security.....	2-86
2.26.1	Overview of Secure Mode.....	2-86
2.26.2	Setting the Console Password.....	2-87
2.26.3	Setting the Console to Secure Mode	2-89
2.26.4	Turning Off Security During a Console Session.....	2-91
2.26.5	Returning to User Mode	2-93
2.27	Updating Firmware	2-94

Chapter 3 AlphaBIOS Console

3.1	Starting AlphaBIOS.....	3-2
3.2	Running a Maintenance Program.....	3-4

Chapter 4 RMC Commands

4.1	RMC Commands List.....	4-2
4.2	clear alert	4-4
4.3	clear port.....	4-5
4.4	disable alert.....	4-6
4.5	disable remote.....	4-7
4.6	enable alert	4-8
4.7	enable remote.....	4-9
4.8	env	4-10
4.9	halt in.....	4-12
4.10	halt out.....	4-13
4.11	hangup	4-14
4.12	help or ?.....	4-15
4.13	power off.....	4-16
4.14	power on.....	4-17
4.15	quit	4-18
4.16	reset.....	4-19
4.17	send alert	4-20
4.18	set alert	4-21
4.19	set com1_mode	4-22
4.20	set dial.....	4-24
4.21	set escape	4-25
4.22	set init	4-26
4.23	set logout.....	4-27
4.24	set password.....	4-28
4.25	set user.....	4-29
4.26	status.....	4-30

Index

Examples

2-1	Help (or Man).....	2-10
2-2	Show Config	2-12
2-3	Show Device	2-17

2-4	Show Fru.....	2-18
2-5	Show Error.....	2-21
2-6	Show Memory.....	2-22
2-7	Show PAL.....	2-23
2-8	Show Power.....	2-24
2-9	Show Version	2-25
2-10	OpenVMS Boot.....	2-26
2-11	Prcache.....	2-28
2-12	Test.....	2-30
2-13	Crash.....	2-32
2-14	Continue.....	2-34
2-15	More	2-35
2-16	Init.....	2-36
2-17	Lpinit.....	2-38
2-18	AlphaBIOS	2-39
2-19	Editing the Nvram Script	2-40
2-20	Clearing the Nvram Script.....	2-40
2-21	Entering RMC from a VGA Monitor	2-42
2-22	Set <i>envar</i> and Show <i>envar</i>	2-44
2-23	Set Password.....	2-87
2-24	Set Secure	2-89
2-25	Login	2-91
2-26	Clear Password	2-93

Figures

3-1	AlphaBIOS Startup Screen.....	3-2
3-2	Utilities Menu	3-3
3-3	Run Maintenance Program Dialog Box	3-4

Tables

2-1	Summary of SRM Console Commands	2-4
2-2	Notation Formats for SRM Console Commands.....	2-6
2-3	Special Characters for SRM Console	2-7
2-4	Management Tasks and Related SRM Commands.....	2-9
2-5	Correspondence Between Logical and Physical PCI Slots.....	2-16
2-6	Device Naming Conventions	2-17
2-7	Bit Assignments for Error Field.....	2-20
2-8	Environment Variable Summary.....	2-46

2-9	OpenVMS Boot Flag Settings	2-55
4-1	Status Command Fields.....	4-31

Preface

Intended Audience

This manual is for managers and operators of *ES40* systems.

Document Structure

This manual uses a structured documentation design. Topics are organized into small sections, usually consisting of two facing pages. Most topics begin with an abstract that provides an overview of the section, followed by an illustration or example. The facing page contains descriptions, procedures, and syntax definitions.

This manual has four chapters.

- **Chapter 1, Console Program Overview**, gives an overview of the console interfaces that underlie the interaction between the server hardware and the supported operating systems.
- **Chapter 2, SRM Console**, describes the SRM console interface for systems running Tru64 UNIX, Linux, or OpenVMS.
- **Chapter 3, AlphaBIOS Console**, describes how to run utilities from the enhanced BIOS graphical interface for *Alpha* systems.
- **Chapter 4, RMC Commands**, describes the command set for the remote management console (RMC).

Documentation Titles

Table 1 ES40 Documentation

Title	Order Number
User Documentation Kit	QA-6E88A-G8
Owner's Guide	EK-ES240-UG
User Interface Guide	EK-ES240-UI
Tower and Pedestal Basic Installation	EK-ES240-PD
Release Notes	
Documentation CD (6 languages)	EK-ES240-RN AG-RF9HB-BE
Maintenance Kit	QZ-01BAB-GZ
Service Guide	EK-ES240-SV
Service Guide HTML CD	AG-RKAKA-BE
Illustrated Parts Breakdown	EK-ES240-IP
Loose Piece Items	
Rackmount Installation Guide	EK-ES240-RG
Rackmount Installation Template	EK-ES4RM-TP

Support Resources

Support resources for this system are available on the Internet, including a supported options list, firmware updates, and patches.

<http://www.digital.com/alphaserver/es40/es40.html>

Preface

Intended Audience

This manual is for managers and operators of *ES40* systems.

Document Structure

This manual uses a structured documentation design. Topics are organized into small sections, usually consisting of two facing pages. Most topics begin with an abstract that provides an overview of the section, followed by an illustration or example. The facing page contains descriptions, procedures, and syntax definitions.

This manual has eight chapters.

- **Chapter 1, System Overview**, gives an overview of the system and describes the components.
- **Chapter 2, Operation**, gives basic operating instructions on powering up and configuring the machine.
- **Chapter 3, Booting and Installing an Operating System**, describes how to boot a supported operating system and how to switch from one operating system to another.
- **Chapter 4, Using the Remote Management Console**, describes the function and operation of the integrated remote management console.
- **Chapter 5, Installing and Configuring Components**, shows how to install components such as memory DIMMs and CPUs.
- **Chapter 6, Updating Firmware**, describes how to update to a later version of system firmware.
- **Chapter 7, Troubleshooting**, gives basic troubleshooting procedures.
- **Chapter 8, Specifications**, gives system specifications.

Documentation Titles

Table 1 ES40 Documentation

Title	Order Number
User Documentation Kit	QA-6E88A-G8
Owner's Guide	EK-ES240-UG
User Interface Guide	EK-ES240-UI
Tower and Pedestal Basic	EK-ES240-PD
Installation	
Release Notes	EK-ES240-RN
Documentation CD (6 languages)	AG-RF9HA-BE
Maintenance Kit	QZ-01BAB-GZ
Service Guide	EK-ES240-SV
Service Guide HTML CD	AG-RKAKA-BE
Illustrated Parts Breakdown	EK-ES240-IP
Loose Piece Items	
Rackmount Installation Guide	EK-ES240-RG
Rackmount Installation Template	EK-ES4RM-TP

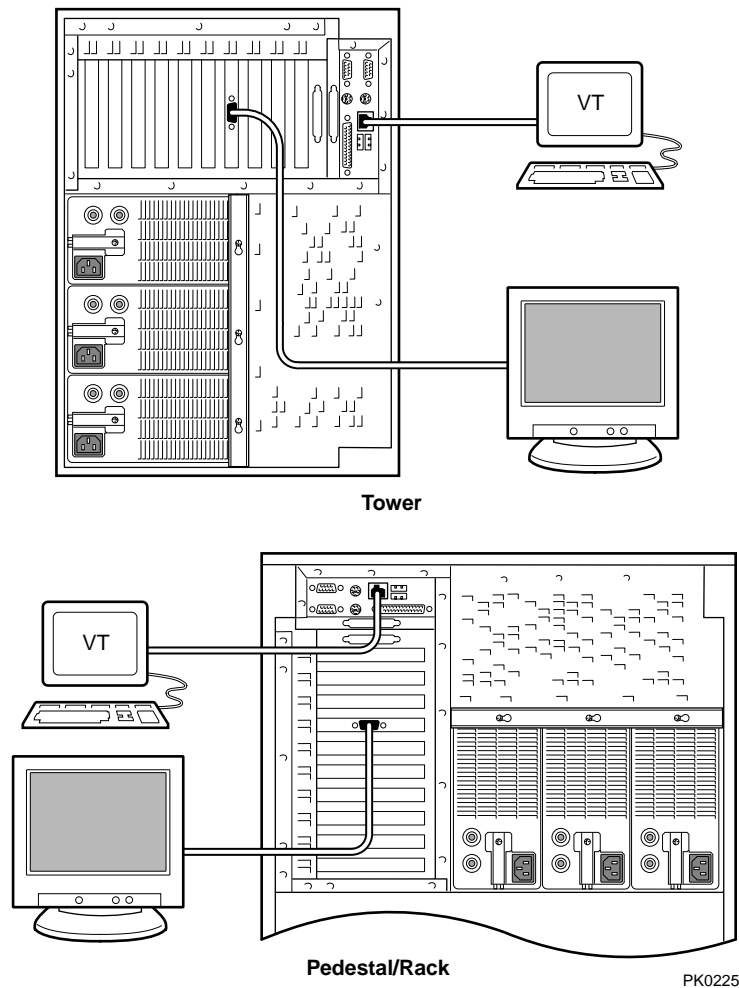
Support Resources

Support resources for this system are available on the Internet, including a supported options list, firmware updates, and patches.

<http://www.digital.com/alphaserver/es40/es40.html>

1.2 Console Terminal

The console terminal that displays the SRM user interface can be either a serial terminal (VT320 or higher, or equivalent) or a VGA monitor.



1.3 Console Mode

When the operating system is halted, the system enters console mode. In console mode, the system operates under the control of a console program and the commands entered or selected by the operator at the console terminal.

1.3.1 SRM Console

SRM (named for the *Alpha System Reference Manual*) is a command-line interface that supports the Tru64 UNIX, Linux, and OpenVMS operating systems.

SRM is a UNIX style shell that provides a set of commands and operators and a script-writing facility. The SRM user interface is used to configure and boot the UNIX or OpenVMS operating system, test the system hardware, and run firmware diagnostics. The SRM console is described in Chapter 2.

1.3.2 AlphaBIOS Console

AlphaBIOS is an enhanced BIOS graphical user interface for Compaq Alpha platforms. AlphaBIOS is used to run AlphaBIOS-compliant utilities.

One such utility is the RAID standalone configuration utility (RCU). The RCU is used to set up the disk drives and logical units on systems that have the optional StorageWorks RAID Array Subsystem. For instructions on using the RCU, refer to the documentation included in your RAID kit. Chapter 3 describes how to run utilities using AlphaBIOS.

1.3.3 Remote Management Console

The remote management console (RMC) is used for either local or remote system management. Through RMC, you can enter commands that emulate the control panel functions and commands that monitor system environmental conditions. The remote management console also provides configuration and error log functionality.

The RMC is described in Chapter 4 of the *ES40 Owner's Guide*. Chapter 4 of this book gives an RMC commands reference.

Chapter 2

SRM Console

The SRM user interface is the command-line interface that allows you to configure and boot the Tru64 UNIX, Linux, or OpenVMS operating system and verify the configuration of devices.

This chapter describes typical functions performed from the SRM console and the commands and environment variables used for these functions. Key sections in this chapter are:

- Invoking the SRM Console
- SRM Command Overview
- Displaying System Information
- Booting the Operating System
- Configuring the System
- Testing the System
- Forcing a System Crash Dump
- Reading a File
- Initializing the System
- Loading the AlphaBIOS Console
- Creating a Power-Up Script
- Entering the RMC from the Local VGA Monitor
- Setting and Viewing Environment Variables
- Ensuring Console Security
- Updating Firmware

2.1 Invoking the SRM Console

You can invoke the SRM console at power-up or restart, after a system failure, or from RMC. Once you invoke SRM, you enter commands at the console prompt, P00>>>.

Invoking SRM from Tru64 UNIX, Linux, or OpenVMS

The SRM console is invoked automatically at power-up or after a reset or failure. The **auto_action** environment variable is set by default to **halt**, which causes the system to stop in the SRM console.

If the operating system is running, invoke the SRM console by shutting down the operating system. Follow the shutdown procedure described in your operating system documentation.

You can also force entry to the SRM console if the **auto_action** environment variable is set to **boot** or **reset**. To force entry, press the Halt button on the control panel.

CAUTION: *A forced halt interrupts the operating system. Applications that are running may lose data.*

Invoking SRM from AlphaBIOS

To invoke the SRM console from AlphaBIOS, press the Reset button.

Invoking SRM from RMC

To invoke the SRM console from the remote management console, issue the following commands:

```
RMC> halt in  
RMC> reset
```

These commands are equivalent to pressing the Halt button on the control panel and then pressing the Reset button. See the *ES40 Owner's Guide* for details on the remote management console. See Chapter 4 of this book for a description of the RMC commands.

To return to operating system mode, unlatch the Halt button or issue the RMC **halt out** command, and boot the operating system.

2.2 SRM Command Overview

Table 2-1 summarizes alphabetically the most frequently used SRM console commands; Table 2-2 gives the command notation formats; and Table 2-3 shows special characters used on the command line.

Table 2-1 Summary of SRM Console Commands

Command	Function
alphabios	Loads and starts the AlphaBIOS console.
boot	Loads and starts the operating system.
continue	Resumes program execution on the specified processor or on the primary processor if none is specified.
crash	Forces a crash dump at the operating system level.
edit	Invokes the console line editor on a RAM script or on the user power-up script, "nvram," which is always invoked during the power-up sequence.
help (or man) command	Displays information about the specified console command.
init	Resets the SRM console and reinitializes the hardware.
lpinit or galaxy	Used in an OpenVMS Galaxy environment. Initializes the hardware resources into zero, one, or two partitions.
more [filename]	Displays a file one screen at a time.
prcache	Initializes and displays the status of the PCI NVRAM.
rmc	Invokes the remote management console from the local VGA monitor.
set envvar	Sets or modifies the value of an environment variable.
show envvar	Displays the state of the specified environment variable.
show config	Displays the logical configuration at the last system initialization.

Table 2-1 Summary of SRM Console Commands (Continued)

Command	Function
show device	Displays a list of controllers and bootable devices in the system.
show error	Reports errors logged in the EEPROMs.
show fru	Displays the physical configuration of all field-replaceable units (FRUs).
show memory	Displays information about system memory.
show pal	Displays the versions of UNIX and OpenVMS PALcode.
show power	Displays information about system environmental characteristics, including power supplies, system fans, CPU fans, and temperature.
show version	Displays the version of the SRM console program installed on the system.
test	Verifies the configuration of the devices in the system.

NOTE: *Diagnostic commands are documented in the ES40 Service Guide.*

Table 2–2 Notation Formats for SRM Console Commands

Attribute	Conditions
Length	Up to 255 characters, not including the terminating carriage return or any characters deleted as the command is entered. To enter a command longer than 80 characters, use the backslash character for line continuation (see Table 2–3).
Case	Upper- or lowercase characters can be used for input. Characters are displayed in the case in which they are entered.
Abbreviation	Only by dropping characters from the end of words. You must enter the minimum number of characters to identify the keyword unambiguously. Abbreviation of environment variables is allowed with the show command.
Options	You can use command options, to modify the environment, after the command keyword or after any symbol or number in the command. See individual command descriptions for examples.
Numbers	Most numbers in console commands are in decimal notation.
No characters	A command line with no characters is a null command. The console program takes no action and does not issue an error message; it returns the console prompt. The console supports command-line recall and editing.
Spaces or tabs	Multiple adjacent spaces and tabs are compressed and treated as a single space. Leading and trailing spaces are ignored.

Table 2–3 Special Characters for SRM Console

Character	Function
Return or Enter	Terminates a command line. No action is taken on a command until it is terminated. If no characters are entered and this key is pressed, the console just redisplay the prompt.
Backslash (\)	Continues a command on the next line. Must be the last character on the line to be continued.
Delete	Deletes the previous character.
Ctrl/A	Toggles between insert and overstrike modes. The default is overstrike.
Ctrl/B or up-arrow	Recalls previous command or commands. The last 16 commands are stored in the recall buffer.
Ctrl/C	Terminates the process that is running. Clears Ctrl/S; resumes output suspended by Ctrl/O. When entered as part of a command line, deletes the current line. Ctrl/C has no effect as part of a binary data stream.
Left-arrow	Moves the cursor left one position.
Ctrl/E	Moves the cursor to the end of the line.
Ctrl/F or right-arrow	Moves the cursor right one position.
Ctrl/H	Moves the cursor to the beginning of the line.
Backspace	Deletes ones character.
Ctrl/J	Deletes the previous word.
Ctrl/O	Stops output to the console terminal for the current command. Toggles between enable and disable. The output can be reenabled by other means as well: when the console prompts for a command, issues an error message, or enters program mode, or when Ctrl/P is entered.

Table 2-3 Special Characters for SRM Console (Continued)

Character	Function
Ctrl/Q	Resumes output to the console terminal that was suspended by Ctrl/S.
Ctrl/R	Redisplays the current line. Deleted characters are omitted. This command is useful for hardcopy terminals.
Ctrl/S	Suspends output to the console terminal until Ctrl/Q is entered. Cleared by Ctrl/C.
Ctrl/U	Deletes the current line.
*	Wildcarding for commands such as show .
" "	Double quotes enable you to denote a string for environment variable assignment.
#	Specifies that all text between it and the end of the line is a comment. Control characters are not considered part of a comment.

2.3 Management Tasks Performed from SRM

This section lists system management tasks and the related SRM commands.

Table 2-4 Management Tasks and Related SRM Commands

Task	Commands
Get help on SRM commands	help or man
Start AlphaBIOS from SRM	alphabios
Display the item indicated (logical configuration, boot devices, FRUs, memory, PALcode version, power supplies and sensors, and SRM version)	show (config, device, error, fru, memory, pal, power, version)
Boot OpenVMS, Tru64 UNIX, or Linux	boot
Initialize (reset) console firmware	init
Initialize the hardware resources in an OpenVMS Galaxy environment	lpinit or galaxy
Set and view environment variables	set <i>envar</i> show <i>envar</i>
Edit a script	edit
Read a file	more
Force a crash dump	crash
Resume program execution	continue
Initialize and display status of PCI NVRAM	prcache
Run RMC from the local VGA monitor	rmc
Verify the devices in the system	test

2.4 Getting Help on SRM Commands

The `help` (or `man`) command displays basic information about a command.

Example 2-1 Help (or Man)

```
P00>>> help set
NAME
    set
FUNCTION
    Set or modify the value of an environment variable.
SYNOPSIS
    set <envar> <value>
        [-integer] [-string]
        where
<envar>={auto_action,bootdef_dev,boot_file,boot_osflags,...}
```

The **help** (or **man**) command displays basic information about the use of console commands when the system is in console mode. The syntax is:

help (or **man**) [*command* . . .]

command . . . Command or topic for which help is requested. The options are:

<i>none</i>	Displays the complete list of commands for which you can receive help.
<i>command_name</i>	Displays information about the console command.
<i>argument_string</i> (such as "sh")	Displays information about all commands that begin with that string.

2.5 Displaying the Logical Configuration

Use the show config command to display the logical configuration of the system. For the physical configuration, see the show fru command (Section 2.7).

Example 2-2 Show Config

```
P00>>>sh config

                                Compaq Computer Corporation
                                Compaq AlphaServer ES40

Firmware
SRM Console:    V5.6-102
ARC Console:    v5.70
PALcode:        OpenVMS PALcode V1.69-2, Tru64 UNIX PALcode V1.62-1
Serial Rom:     V2.5-F
RMC Rom:        V1.1
RMC Flash Rom:  V2.2

Processors
CPU 0           Alpha EV67 pass 2.2.3 667 MHz 8MB Bcache
CPU 1           Alpha EV67 pass 2.2.3 667 MHz 8MB Bcache
CPU 2           Alpha EV67 pass 2.2.3 667 MHz 8MB Bcache
CPU 3           Alpha EV67 pass 2.2.3 667 MHz 8MB Bcache

Core Logic
Cchip           DECchip 21272-CA Rev 9(C4)
Dchip           DECchip 21272-DA Rev 2
Pchip 0         DECchip 21272-EA Rev 2
Pchip 1         DECchip 21272-EA Rev 2
TIG             Rev 10

Memory
  Array      Size      Base Address      Intlv Mode
  -----
    0         256Mb      0000000060000000      2-Way
    1         512Mb      0000000040000000      2-Way
    2         256Mb      0000000070000000      2-Way
    3        1024Mb      0000000000000000      2-Way

    2048 MB of System Memory
```

- ❶ **Firmware.** Version numbers of the SRM console, AlphaBIOS (ARC) console, PALcode, serial ROM, RMC ROM and RMC flash ROM.
- ❷ **Processors.** Processors present, processor version and clock speed, and amount of backup cache.
- ❸ **Core logic.** Version numbers of the chips on the system motherboard.
- ❹ **Memory.** Memory arrays and memory size.

Continued on next page

Example 2-2 Show Config (Continued)

```

Slot  Option                               Hose 0, Bus 0, PCI                               ⑤
  2    ELSA GLoria Synergy
  7    Acer Labs M1543C
 15    Acer Labs M1543C IDE      dqa.0.0.15.0      Bridge to Bus 1, ISA
                                   dqb.0.1.15.0
                                   dqa0.0.0.15.0      TOSHIBA CD-ROM XM-6302B

 19    Acer Labs M1543C USB
        Option                               Hose 0, Bus 1, ISA
        Floppy                             dva0.0.0.1000.0

Slot  Option                               Hose 1, Bus 0, PCI
  4    DE500-BA Network Con      ewa0.0.0.4.1      00-00-F8-09-90-FF
  6    DECchip 21152-AA                               Bridge to Bus 2, PCI

Slot  Option                               Hose 1, Bus 2, PCI
  0    NCR 53C875                pka0.7.0.2000.1   SCSI Bus ID 7
                                   dka0.0.0.2000.1   RZ2DD-LS
                                   dka100.1.0.2000.1  RZ2DD-LS
                                   dka200.2.0.2000.1  RZ1CB-CS
  1    NCR 53C875                pkb0.7.0.2001.1   SCSI Bus ID 7
  2    DE500-AA Network Con      ewb0.0.0.2002.1   00-06-2B-00-25-5B
P00>>>

```


⑤ PCI bus information.

The “Slot” column lists the logical slots seen by the system. They are not the physical slots into which devices are installed. See Table 2–5 for the correspondence between logical slots and physical slots.

The NCR 53C896 on Hose 0, Bus 0 is a dual-channel Ultra2 SCSI multifunction controller. Two controllers reside on the same chip. They are shown as 2/0 and 2/1. The first number is the logical slot, and the second is the function.

The Acer Labs bridge chip, which is located in PCI logical slot 7, has two built-in IDE controllers. The CD-ROM is on the first controller.

NOTE: The *naming of devices (for example, dqa.0.0.15.0) follows the conventions described in Table 2–6.*

In Example 2–2, the following devices are present:

Hose 0, Bus 0, PCI

Slot 2/0	SCSI controller
Slot 2/1	SCSI controller
Slot 4	VGA controller
Slot 7	PCI to ISA bridge chip
Slot 15	IDE controller and CD-ROM drive
Slot 19	Universal serial bus (USB) controller

Hose 0, Bus 1, ISA

Diskette drive

Hose 1, Bus 0, PCI

Slot 1	SCSI controller and drives
Slot 3	SCSI controller and drives
Slot 4	Ethernet controller
Slot 6	PCI-to-PCI bridge chip to Bus 2

Hose 1, Bus 2, PCI

Slot 0	SCSI controller
Slot 1	SCSI controller
Slot 2	Ethernet controller

Continued on next page

Table 2-5 Correspondence Between Logical and Physical PCI Slots

Physical Slot	Logical Slot	PCI 0
1	1	Device
2	2	Device
3	3	Device
4	4	Device
Physical Slot	Logical Slot	PCI 1
5	1	Device
6	2	Device
7	3	Device
8	4	Device
9	5	Device
10	6	Device

NOTE: *PCI 0 and PCI 1 correspond to Hose 0 and Hose 1 in the logical configuration.*

2.6 Displaying the Bootable Devices

Use the show device command to display the bootable devices. DK = SCSI drive; DQ = IDE drive; DV = diskette drive; EI or EW = Ethernet controller; PK = SCSI controller.

Example 2-3 Show Device

```
P00>>> show device
dka0.0.0.1.1          DKA0          RZ2DD-LS  0306
dka100.1.0.1.1        DKA100         RZ2DD-LS  0306
dka200.2.0.1.1        DKA200         RZ1CB-CS  0844
dkb0.0.0.3.1          DKB0           RZ25      0900
dqa0.0.0.15.0         DQA0           TOSHIBA CD-ROM XM-6302B 1012
dva0.0.0.1000.0       DVA0
ewa0.0.0.4.1          EWA0           00-00-F8-09-90-FF
ewb0.0.0.2002.1       EWB0           00-06-2B-00-25-5B
pka0.7.0.1.1          PKA0           SCSI Bus ID 7
pkb0.7.0.3.1          PKB0           SCSI Bus ID 7
pkc0.7.0.2000.1       PKC0           SCSI Bus ID 7
pkd0.7.0.2001.1       PKD0           SCSI Bus ID 7
```

Table 2-6 Device Naming Conventions

Category	Description
dq Driver ID	Two-letter designator of port or class driver
	dk SCSI drive or CD ew Ethernet port
	dq IDE CD-ROM fw FDDI device
	dr RAID set device mk SCSI tape
	du DSSI disk mu DSSI tape
	dv Diskette drive pk SCSI port
	ei Ethernet port pu DSSI port
a Storage adapter ID	One-letter designator of storage adapter (a, b, c...).
0 Device unit number	Unique number (MSCP unit number). SCSI unit numbers are forced to 100 X node ID.
0 Bus node number	Bus node ID.
0 Channel number	Used for multi-channel devices.
15 Logical slot num.	Corresponds to PCI slot number (see Table 2-5).
0 Hose number	0 — PCI 0 1 — PCI 1

2.7 Displaying the System FRUs

Use the show fru command to view the physical configuration of FRUs.
Use the show error command to display FRUs with errors.

Example 2-4 Show Fru

```
P00>>> show fru
```

①	②	③	④	⑤	⑥
FRUname	E	Part#	Serial#	Misc.	Other
SMB0	00	54-25385-01.E01	AY94412345		
SMB0.CPU0	00	54-30158-A5	NI90260078		
SMB0.CPU1	00	54-30158-A5	NI90260073		
SMB0.CPU2	00	54-30158-A5	NI90260056		
SMB0.CPU3	00	54-30158-A5	NI90260071		
SMB0.MMB0	00	54-25582-01.B02	AY90112345		
SMB0.MMB0.DIM1	00	54-24941-EA.A01CPQ	NI90202001		
SMB0.MMB0.DIM2	00	54-24941-EA.A01CPQ	NI90200102		
SMB0.MMB0.DIM3	00	54-24941-EA.A01CPQ	NI90200103		
SMB0.MMB0.DIM4	00	54-24941-EA.A01CPQ	NI90200104		
SMB0.MMB0.DIM5	00	54-24941-EA.A01CPQ	NI90202005		
SMB0.MMB0.DIM6	00	54-24941-EA.A01CPQ	NI90202006		
SMB0.MMB1	00	54-25582-01.B02	AY90112301		
SMB0.MMB1.DIM1	00	54-25053-BA.A01CPQ	NI90112341		
SMB0.MMB1.DIM2	00	54-25053-BA.A01CPQ	NI90112342		
SMB0.MMB1.DIM3	00	54-25053-BA.A01CPQ	NI90112343		
SMB0.MMB1.DIM4	00	54-25053-BA.A01CPQ	NI90112344		
SMB0.MMB1.DIM5	00	54-25053-BA.A01CPQ	NI90112345		
SMB0.MMB1.DIM6	00	54-25053-BA.A01CPQ	AY80112346		
SMB0.MMB2	00	54-25582-01.B02	AY80012302		
SMB0.MMB2.DIM1	00	54-25053-BA.A01CPQ	NI90112331		
SMB0.MMB2.DIM2	00	54-25053-BA.A01CPQ	AY80112332		
SMB0.MMB2.DIM3	00	54-25053-BA.A01CPQ	AY80112333		
SMB0.MMB2.DIM4	00	54-25053-BA.A01CPQ	AY80112334		
SMB0.MMB2.DIM5	00	54-25053-BA.A01CPQ	AY80112335		
SMB0.MMB2.DIM6	00	54-25053-BA.A01CPQ	AY80112336		
SMB0.MMB3	00	54-25582-01.B02	AY90112303		
SMB0.MMB3.DIM1	00	54-25053-BA.A01CPQ	AY80112341		
SMB0.MMB3.DIM2	00	54-25053-BA.A01CPQ	AY80112342		
SMB0.MMB3.DIM3	00	54-25053-BA.A01CPQ	AY80112343		
SMB0.MMB3.DIM4	00	54-25053-BA.A01CPQ	AY80112344		
SMB0.MMB3.DIM5	00	54-25053-BA.A01CPQ	AY80112345		
SMB0.MMB3.DIM6	00	54-25053-BA.A01CPQ	AY80112346		
SMB0.CPB0	00	54-25573-01	AY80100999		
SMB0.CPB0.PCI4	00	ELSA GLoria Synergy			
SMB0.CPB0.PCI5	00	NCR 53C895			
SMB0.CPB0.PCIA	00	DE500-BA Network Cont			
SMB0.CPB0.SBM0	00	-	-		
PWR0	00	30-49448-01.A02	2P90700557	API-7850	
PWR1	00	30-49448-01.A02	2P90700558	API-7850	
FAN1	00	70-40073-01	-	Fan	
FAN2	00	70-40073-01	-	Fan	
FAN3	00	70-40072-01	-	Fan	
FAN4	00	70-40071-01	-	Fan	

FAN5	00 70-40073-02	-	Fan
FAN6	00 70-40074-01	-	Fan
JIO0	00 54-25575-01	-	Junk I/O
OCP0	00 70-33894-0x	-	OCP

P00>>>

- ❶ **FRUname** The FRU name recognized by the SRM console. The name also indicates the location of that FRU in the physical hierarchy.

SMB = system motherboard; CPU = CPUs; MMB = memory motherboard; DIM = DIMMs; CPB = PCI backplane; PCI = PCI option; SBM = SCSI backplane; PWR = power supply; FAN = fans; JIO= I/O connector module (junk I/O).
- ❷ **E** Error field. Indicates whether the FRU has any errors logged against it. FRUs without errors show 00 (hex). FRUs with errors have a non-zero value that represents a bit mask of possible errors. See Table 2-7.
- ❸ **Part #** The part number of the FRU in ASCII, either a Compaq part number or a vendor part number.
- ❹ **Serial #** The serial number. For Compaq FRUs, the serial number has the form XXYWWNNNNN.
XX = manufacturing location code
YWW = year and week
NNNNN = sequence number. For vendor FRUs, the 4-byte sequence number is displayed in hex.
- ❺ **Misc.** Miscellaneous information about the FRUs. For Compaq FRUs, a model name, number, or the common name for the entry in the Part # field. For vendor FRUs, the manufacturer's name.
- ❻ **Other** Optional data. For Compaq FRUs, the Compaq part alias number (if one exists). For vendor FRUs, the year and week of manufacture.

Table 2–7 Bit Assignments for Error Field

Bit	Meaning
Bit 0 is 1	Failure
Bit 1 is 1	TDD error has been logged
Bit 2 is 1	At least one SDD error has been logged
Bit 3 is 1	FRU EEPROM is unreadable
Bit 4 is 1	Checksum failure on bytes 0-62
Bit 5 is 1	Checksum failure on bytes 64-126
Bit 6 is 1	Checksum failure on bytes 128-254
Bit 7 is 1	FRU's system serial does not match system's

NOTE: *Contact your service provider if the E (error) field shows any of these errors.*

2.8 Displaying FRUs with Errors

The `show error` command displays FRUs that have errors logged to the serial control bus EEPROMs.

Example 2-5 Show Error

```
P00>>> show error
SMB0      TDD - Type: 1 Test: 1 SubTest: 1 Error: 1
SMB0      SDD - Type: 4 LastLog: 1 Overwrite: 0
P00>>>
```

The output of the **show error** command is based on information logged to the serial control bus EEPROMs. Both the operating system and the ROM-based diagnostics log errors to the EEPROMs. This functionality allows service providers to generate an error log from the console environment.

The syntax is:

show error

If no errors are logged, nothing is displayed and you are returned to the SRM console prompt.

Example 2-5 shows errors reported on the system motherboard (SMB0). Contact your service provider if errors are displayed.

2.9 Displaying the Memory Configuration

Use the **show memory** command to view the total memory size and location.

Example 2-6 Show Memory

```
P00>>>show memory
```

Array	Size	Base Address	Intlv Mode
0	256Mb	0000000060000000	2-Way
1	512Mb	0000000040000000	2-Way
2	256Mb	0000000070000000	2-Way
3	1024Mb	0000000000000000	2-Way

2048 MB of System Memory

The **show memory** display corresponds to the memory array configuration shown in the *ES40 Owner's Guide*. The display does not indicate the number of DIMMs or their size. Thus, in Example 2-6, Array 3 could consist of two sets of 128 MB DIMMs (eight DIMMs) or one set of 256 MB DIMMs (four DIMMs). Either combination provides 1024 MB of memory.

Use the **show fru** command to display all the DIMMs in the system and their locations.

2.10 Displaying the PAL Version

Use the `show pal` command to display the PALcode version.

Example 2-7 Show PAL

```
P00>>> show pal
pal      OpenVMS PALcode V1.69-2, Tru64 UNIX PALcode V1.62-1
P00>>>
```

The **`show pal`** command displays the versions of UNIX and OpenVMS PALcode. PALcode is the Alpha Privileged Architecture Library code, written to support Alpha processors. It implements architecturally defined processor behavior.

The syntax is:

`show pal`

2.11 Displaying the Power Status

Use the show power command to display the status of power supplies, fans, and system temperature. If you are not able to access SRM, invoke RMC and issue the env command.

Example 2-8 Show Power

```
P00>>> show power

Power Supply 0      Status      ❶
Power Supply 1      Good
Power Supply 2      Not Available
System Fan 1        Good      ❷
System Fan 2        Good
System Fan 3        Bad
System Fan 4        Good
System Fan 5        Good
System Fan 6        Good
CPU 0 Temperature   Warning    ❸
CPU 1 Temperature   Good
CPU 2 Temperature   Good
CPU 3 Temperature   Good
Zone 0 Temperature   Good      ❹
Zone 1 Temperature   Good
Zone 2 Temperature   Good
P00>>>
```

- ❶ Power supplies. Power supply 2 is not installed.
- ❷ System fans. Fan 3 is not working.
- ❸ Temperature sensors on CPUs. CPU 0 is above threshold.
- ❹ Temperature sensors on PCI backplane.

The syntax is:

show power

2.12 Displaying the SRM Console Version

Use the `show version` command to display the version of the SRM console that is installed.

Example 2-9 Show Version

```
P00>>> show version
version                V5.6-102 Nov 29 1999 10:47:31
P00>>>
```

The **`show version`** command displays the version of the SRM console program that is installed on the system.

The syntax is:

`show version`

2.13 Booting an Operating System

The **boot** command boots the Tru64 UNIX, Linux, or OpenVMS operating system. You can specify a boot device, operating system-specific boot information (boot flags), and an Ethernet protocol for network boots. You can also specify whether the boot program should halt and remain in console mode.

Example 2-10 OpenVMS Boot

```
P00>>> boot dkb0
(boot dkb0.0.0.2.1 -flags 0)
block 0 of dkb0.0.0.2.1 is a valid boot block
reading 1002 blocks from dkb0.0.0.2.1
bootstrap code read in
base = 200000, image_start = 0, image_bytes = 7d400
initializing HWRPB at 2000
initializing page table at 1f2000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
```

OpenVMS (TM) Operating System, Version 7.2-1

The **boot** command initializes the processor, loads a program image from the specified boot device, and transfers control to that image. If you do not specify a boot device in the command line, the default boot device is used. The default boot device is determined by the value of the **bootdef_dev** environment variable, described in Section 2.25.2.

If you specify a list of boot devices, a bootstrap is attempted from each device in order. Then control passes to the first successfully booted image. In a list, always enter network devices last, because network bootstraps terminate only if a fatal error occurs or when an image is successfully loaded.

The syntax is:

```
boot [-file filename] [-flags [value]] [-halt] [-protocols enet_protocol]
[boot_dev]
```

-file
filename Specifies the name of a file to load into the system. Use the **set boot_file** command to set a default boot file (Section 2.25.3).

NOTE: *For booting from Ethernet, the filename is limited by the MOP V3 load protocol to 15 characters. The MOP protocol is used with OpenVMS systems.*

-flags
[value] Provides additional operating system-specific boot information. In UNIX, specifies boot flags. In OpenVMS, specifies system root number and boot flags. These values are passed to the operating system for interpretation. Preset default boot flag values are 0,0. Use the **set boot_osflags** command to change the default boot flag values. See Section 2.25.4.

-halt Forces the bootstrap operation to halt and invoke the console program. The console is invoked after the bootstrap image is loaded and page tables and other data structures are set up. Console device drivers are not shut down. Transfer control to the bootstrap image by entering the **continue** command.

-protocols
enet_protocol Specifies the Ethernet protocol to be used for the network boot. Either **mop** (for OpenVMS) or **bootp** (for UNIX) may be specified. Use the **set ew*0_protocols** or **ei*0_protocols** command to set a default network boot protocol. See Section 2.25.14.

boot_dev A device path or list of devices from which the console program attempts to boot, or a saved boot specification in the form of an environment variable. Use the **set bootdef_dev** command to set a default boot device. See Section 2.25.2.

NOTE: *Entering values for boot flags, the boot device name, or Ethernet protocol on the **boot** command overrides the current default value for the current boot request, but does not change the corresponding environment variable. For example, if you have defined a value for **boot_osflags** and you specify the **-flags** argument on the **boot** command line, the **-flags** argument takes precedence for that boot session.*

2.14 Configuring a PCI NVRAM Module

The `prcache` command is used to support Tru64 UNIX systems equipped with a Prestoserve PCI NVRAM module.

Example 2-11 Prcache

```
P00>>> prcache -f
PCI NVRAM Disk Cache: passed
Size:      4MB
PCI Memory Address: 40000000
System ID: 12000000
State:     - not valid
Battery Status: good (Charging)
Battery Disconnect Circuit Status: enabled

P00>>> prcache -z
This command will zero the PCI NVRAM Disk Cache
Do you really want to continue [Y/N] ? : y
clearing disk cache
P00>>>
```

The **`prcache`** command, with the options listed below, checks PCI NVRAM configuration and battery status, clears data from the NVRAM module, and disables the NVRAM battery disconnect circuit. This command is used only with UNIX systems. The syntax is:

`prcache -{f,z,u}`

- f** Checks configuration and battery status.
- z** Clears valid data; writes zeros to memory.
- u** Disables the NVRAM battery disconnect circuit.

2.15 Configuring RAID Devices

RAID devices are configured from the AlphaBIOS console.

RAID devices are configured with the RAID standalone configuration utility. See Chapter 3 for information about running AlphaBIOS utility programs.

2.16 Testing the System

The `test` command verifies the configuration of the devices in the system.

Example 2-12 Test

```
P00>>> test
Testing the Memory
Testing the DK* Disks(read only)
No DU* Disks available for testing
No DR* Disks available for testing
Testing the DQ* Disks(read only)
Testing the DF* Disks(read only)
No MK* Tapes available for testing
No MU* Tapes available for testing
Testing the DV* Floppy Disks(read only)
Testing the VGA (Alphanumeric Mode only)
Testing the EWA0 Network
Testing the EWB0 Network
P00>>>
```

The **`test`** command also does a quick test on the system speaker. A beep is emitted as the command starts to run.

The tests are run sequentially, and the status of each subsystem test is displayed to the console terminal as the tests progress. If a particular device is not available to test, a message is displayed. The test script does no destructive testing; that is, it does not write to disk drives.

The syntax is:

`test` [*argument*]

Use the **`-lb`** (loopback) argument for console loopback tests.

To run a complete diagnostic test using the **test** command, the system configuration must include:

- A serial loopback connected to the COM2 port (not included)
- A parallel loopback connected to the parallel port (not included)
- A formatted diskette
- A formatted CD-ROM

The test script tests devices in the following order:

1. Memory tests (one pass)
2. Read-only tests: DK* disks, DR* disks, DQ* disks, DU* disks, MK* tapes, DV* floppy.

NOTE: *You must install media to test disks, tapes, and the floppy drive. Since no write tests are performed, it is safe to test disks and tapes that contain data.*

3. Console loopback tests if **-lb** argument is specified: COM2 serial port and parallel port.
4. VGA console tests: These tests are run only if the console environment variable is set to **serial**. The VGA console test displays rows of the word *compaq*.
5. Network internal loopback tests for EW* networks.

2.17 Forcing a System Crash Dump

The crash command causes an operating system that has hung to crash so that you can capture a crash dump to the selected device.

Example 2-13 Crash

```
P00>>> crash
```

```
CPU 0 restarting
```

```
DUMP: 19837638 blocks available for dumping.
DUMP: 118178 wanted for a partial compressed dump.
DUMP: Allowing 2060017 of the 2064113 available on 0x800001
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Header to 0x800001 at 2064113 (0x1f7ef1)
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Dump to 0x800001: .....: End 0x800001
device string for dump = SCSI 1 1 0 0 0 0 0.
DUMP.prom: dev SCSI 1 1 0 0 0 0 0, block 2178787
DUMP: Header to 0x800001 at 2064113 (0x1f7ef1)
succeeded
```

```
halted CPU 0
```

```
halt code = 5
HALT instruction executed
PC = fffffc0000568704
P00>>>
```

The **crash** command forces an operating system that has stopped responding to crash so that you can capture a crash dump.

The syntax is:

crash

Press the Halt button or use the RMC **halt in** command to invoke the SRM console, then enter the **crash** command to restart the primary CPU and force a crash dump to the selected device.

- See the *OpenVMS Alpha System Dump Analyzer Utility Manual* for information on how to interpret OpenVMS crash dump files.
- See the *Guide to Kernel Debugging* for information on using the Tru64 UNIX Krash Utility.

2.18 Resuming Program Execution

The `continue` command resumes the execution of the operating system on the processor you specify or on the primary processor, if none is specified.

Example 2-14 Continue

```
P00>>> halt
P00>>> continue

continuing CPU
```

Typically, you use the **`continue`** command if you inadvertently halt the system and want to resume operating system mode. The **`continue`** command is valid under the following circumstances:

- If you pressed the Halt button on the control panel or entered the RMC **`halt in`** command. You must unlatch the Halt button or enter the RMC **`halt out`** command before issuing the **`continue`** command.
- If you used the **`-halt`** option with the **`boot`** command
- If you issued a Ctrl/P at the SRM console (OpenVMS systems only)
- If you do not disturb the machine state after the halt by entering other SRM commands

The syntax is:

`continue`

2.19 Reading a File

The `more` command displays a file one screen at a time.

Example 2-15 More

```
P00>>> more el ❶  
*** Error - CPU 1 failed powerup diagnostics ***  
Secondary start error  
.  
.  
.  
  
P00>>> help * | more ❷
```

- ❶ Displays the contents of the SRM console's event log one screen at a time.
- ❷ Displays the contents of online help one screen at a time.

The SRM **more** command is similar to the UNIX **more** command. It is useful for displaying output that scrolls too quickly to be viewed. For example, when you power up the system, the system startup messages scroll, and the messages are written to an event log. When the P00>>> prompt is displayed, you can use the **more el** command to display the contents of the event log file. The syntax is:

more [*file...*]

File is the name of the file to be displayed.

2.20 Initializing the System

The init command resets the SRM console firmware and reinitializes the hardware. Example 2-16 shows an abbreviated example.

Example 2-16 Init

```
OpenVMS PALcode V1.69-2, Tru64 UNIX PALcode V1.62-1
starting console on CPU 0
initialized idle PCB
initializing semaphores
initializing heap
initial heap 200c0
memory low limit = 154000
heap = 200c0, 17fc0
initializing driver structures
initializing idle process PID
initializing file system
initializing hardware
initializing timer data structures
lowering IPL
CPU 0 speed is 667 MHz
create dead_eater
create poll
create timer
create powerup
access NVRAM
Memory size 2048 MB
testing memory
. . .
probe I/O subsystem
probing hose 1, PCI
probing PCI-to-PCI bridge, bus 2
bus 0, slot 4 -- ewa -- DE500-BA Network Controller
bus 2, slot 0 -- pka -- NCR 53C875
bus 2, slot 1 -- pkb -- NCR 53C875
bus 2, slot 2 -- ewb -- DE500-AA Network Controller
probing hose 0, PCI
probing PCI-to-ISA bridge, bus 1
bus 0, slot 2 -- vga -- ELSA GLoria Synergy
bus 0, slot 15 -- dqa -- Acer Labs M1543C IDE
bus 0, slot 15 -- dqb -- Acer Labs M1543C IDE
starting drivers
entering idle loop
initializing keyboard
starting console on CPU 1
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 1 speed is 667 MHz
```

```

create powerup
starting console on CPU 2
initialized idle PCB
initializing idle process PID
lowering IPL
CPU 2 speed is 667 MHz
.
.
.
Memory Testing and Configuration Status
  Array      Size      Base Address      Intlv Mode
  -----
    0         256Mb      0000000060000000      2-Way
    1         512Mb      0000000040000000      2-Way
    2         256Mb      0000000070000000      2-Way
    3        1024Mb      0000000000000000      2-Way

    2048 MB of System Memory
Partition 0, Memory base: 000000000, size: 080000000
initializing GCT/FRU at 1a6000
AlphaServer ES40 Console V5.6-102, built on Dec 2 1999 at 10:47:31

```

The **init** command restarts the SRM console and reinitializes the hardware. This command is similar to performing a reset, but a reset causes full start-up diagnostics to be performed, whereas **init** performs only the SRM diagnostics. The syntax is:

init

After you use the **init** command, the system stops in the SRM console because the **auto_action** environment variable is set by default to **halt**. To cause the system to boot automatically after issuing the **init** command, set the **auto_action** environment variable to **boot** or **restart**.

New values for the following environment variables take effect only after you reset the system by pressing the Reset button or issuing the **init** command:

```

auto_action
console
cpu_enabled
os_type
pk*0_fast
pk*0_host_id
pk*0_soft_term
console_memory_allocation

```

2.21 Initializing the Hardware Resources in an OpenVMS Galaxy

The lpinit command initializes the hardware resources in an OpenVMS Galaxy environment into zero, one, or two partitions. The galaxy command performs the same function. Use lpinit after you have set up the lp* environment variables.

Example 2-17 Lpinit

```
P00>>>sho lp*
lp_count                2
lp_cpu_mask0            5
lp_cpu_mask1            a
lp_io_mask0             1
lp_io_mask1             2
lp_mem_size0            0
lp_mem_size1            0
lp_shared_mem_size      0
P00>>>show auto_action
P00>>>auto_action      halt
P00>>>lpinit
Partition 0: Primary CPU = 0
Partition 1: Primary CPU = 1
LP Configuration Tree = 194000
starting cpu 1 in partition 1 at address 020008001
starting cpu 3 in partition 1 at address 020008001
.
.
.
```

NOTE: If **auto_action** is set to **halt**, you must issue the **lpinit** command every time before booting for the Galaxy features to work. If **auto_action** is set to **boot**, initialization of the Galaxy partitions occurs automatically. For information on Galaxy, see the DS20E-ES40 Remedial Kit available at <http://www.service.digital.com/patches>.

2.22 Loading the AlphaBIOS Console

The `alphabios` command loads and starts the AlphaBIOS console. AlphaBIOS-based utilities, such as the RAID configuration utility, are run from AlphaBIOS.

Example 2-18 AlphaBIOS

```
P00>>> alphabios -g
Loading Arc Firmware From Flash
resetting all I/O buses
Arc Firmware Loaded
```

The syntax is:

alphabios

Options

-g Starts AlphaBIOS on a VGA port. Use this option if the **console** environment variable is set to **serial**, but you want AlphaBIOS to come up on the VGA monitor.

To return to the SRM console, reset the system by pressing the Reset button.

2.23 Creating a Power-Up Script

The system comes with a script (set of commands) named “nvram” that is stored in EEROM. Nvram is a user-created power-up script that is always invoked during the power-up sequence. Use the SRM edit command to create or alter the nvram script.

Example 2-19 Editing the Nvram Script

```
P00>>> edit nvram
editing 'nvram'
0 bytes read in
*10 set mopv3_boot 1
*exit
17 bytes written out to nvram
P00>>>
```

Example 2-20 Clearing the Nvram Script

```
P00>>> edit nvram
editing 'nvram'
20 bytes read in
*10
*exit
0 bytes written out to nvram
P00>>>
```

You can create an nvram script to include any commands you want the system to execute at power-up. You create and edit the nvram script using the SRM **edit** command. With **edit**, lines may be added, overwritten, or deleted. To clear the script, enter line numbers without any text. This deletes the lines.

In Example 2–19 an environment variable called “mop3_boot” is created and set to 1 on each power-up. By default, MOP boots send four MOP V4 requests before defaulting to MOP V3. This user-created environment variable forces the SRM console to bypass MOP V4 requests. This speeds up MOP booting on networks with MOP V3 software. The syntax is:

edit *file*

The *file* is the name of the file to be edited.

The editing commands are:

help	Displays the brief help file.
list	Lists the current file prefixed with line numbers.
renumber	Renumbers the lines of the file in increments of 10.
exit	Leaves the editor and closes the file, saving all changes.
quit	Leaves the editor and closes the file without saving changes.
<i>nn</i>	Deletes line number <i>nn</i> .
<i>nn text</i>	Adds or overwrites line number <i>nn</i> with the specified text.

CAUTION: *Use caution when editing the nvram script. It is possible to disable the system by including an inappropriate command. For example, if you include the **init** command in the script, the system will go into an endless loop.*

*To correct this error, press the Halt button or issue the RMC **halt in** command, then power up or reset the system. When the P00>>> prompt is displayed, edit the nvram script to remove the illegal command.*

2.24 Entering the RMC from the Local VGA Monitor

Use the rmc command to enter the remote management console from a VGA monitor connected to the system. All RMC commands are available and all output is redirected to the VGA monitor.

Example 2-21 Entering RMC from a VGA Monitor

Failing Connection

```
P00>>> rmc
Unable to allocate COM1. Currently in use by: shell
To connect to the Remote Management Console from the
graphics interface the "console" environment variable
must be set to "graphics," and the serial interface cannot
be in use by another console program.
RMC>
```

Successful Connection

```
P00>>> rmc
You are about to connect to the Remote Management Console.
Use the RMC reset command or press the front panel reset
button to disconnect and to reload the SRM console.
Do you really want to continue? [y/(n)] y
Please enter the escape sequence to connect to the Remote
Management Console.
```

Exiting from the Graphics Interface

```
RMC> reset
```

The **rmc** command allows you to invoke the remote management console (RMC) from a VGA monitor connected to the VGA port. The syntax is:

rmc

After entering the **rmc** command, type the default escape sequence to connect to the RMC. The default escape sequence is:

```
^[^[rmc
```

This sequence is equivalent to Ctrl/left bracket, Ctrl/left bracket, rmc. On some keyboards, the escape key functions like the Ctrl/left bracket combination.

The **console** environment variable must be set to **graphics**, and the serial interface cannot be in use by another console program. Once you have connected to RMC, the serial interface is disabled. See Example 2–21 for an example of an unsuccessful connection and a successful connection.

To exit RMC, enter the RMC **reset** command or press the Reset button on the control panel. The reset disconnects the RMC session, resets hardware, and reloads the SRM console from the flash ROM.

See the *ES40 Owner's Guide* for complete information about RMC. See Chapter 4 in this book for a description of the RMC command set.

2.25 Setting and Viewing Environment Variables

Use the **set *envvar*** and **show *envvar*** commands to set and view environment variables.

Example 2-22 Set *envvar* and Show *envvar*

```
P00>>> set bootdef_dev dkb0
P00>>> show bootdef_dev
Bootdef_dev dkb0
```

Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates. Environment variables are set or changed with the **set *envvar*** command. Their values are viewed with the **show *envvar*** command.

set *envvar*

The **set** command sets or modifies the value of an environment variable. It can also be used to create a new environment variable if the name used is unique. Environment variables pass configuration information between the console and the operating system. Their settings determine how the system powers up, boots the operating system, and operates. The syntax is:

set *envvar value*

envvar The name of the environment variable to be modified. See Table 2–8 for a list of environment variables

value The new value of the environment variable.

New values for the following environment variables take effect only after you reset the system by pressing the Reset button or by issuing the **init** command. A reset or **init** is also required when setting up environment variables for an OpenVMS Galaxy configuration. See the DS20E-ES40 Remedial Kit available from <http://www.service.digital.com/patches>.

auto_action
console
cpu_enabled
os_type
pk*0_fast
pk*0_host_id
pk*0_soft_term
console_memory_allocation

show *envvar*

The **show *envvar*** command displays the current value (or setting) of an environment variable. The syntax is:

show *envvar*

envvar The name of the environment variable to be displayed. The **show*** command displays all environment variables.

Table 2–8 summarizes the SRM environment variables. These environment variables are described in the following pages.

Continued on next page

Table 2–8 Environment Variable Summary

Environment Variable	Function
auto_action	Specifies the console's action at power-up, a failure, or a reset.
bootdef_dev	Specifies the default boot device string.
boot_file	Specifies the default file name to be used for booting when no file name is specified by the boot command.
boot_osflags	Specifies the default operating system boot flags.
com1_baud	Sets the baud rate of the internal COM1 serial interface.
com2_baud	Sets the default baud rate of the COM2 serial port.
com1_flow or com2_flow	Specifies the flow control on the serial ports.
com1_mode	Specifies the COM1 data flow paths so that data either flows through the RMC or bypasses it.
com1_modem or com2_modem	Specifies to the operating system whether or not a modem is present.
console	Specifies the device on which power-up output is displayed (serial terminal or VGA monitor).
console_memory_allocation	Determines which memory locations the console will allocate for its private use.
cpu_enabled	Enables or disables a specific secondary CPU.
ei*0_inet_init or ew*0_inet_init	Determines whether the interface's internal Internet database is initialized from nvram or from a network server (by using the bootp protocol).
ei*0_mode or ew*0_mode	Specifies the connection type of the default Ethernet controller.
ei*0_protocols or ew*0_protocols	Specifies network protocols for booting over the Ethernet controller.

Table 2–8 Environment Variable Summary (Continued)

Environment Variable	Function
kbd_hardware_type	Specifies the default console keyboard type.
language	Specifies the console keyboard layout.
lp_count	Specifies the number of Galaxy partitions to create.
lp_cpu_mask <i>n value</i>	Specifies which CPUs are to be assigned to the <i>n</i> Galaxy partition.
lp_io_mask <i>n value</i>	Specifies bitmask of I/O IDs to be included in the <i>n</i> Galaxy partition.
lp_mem_size <i>n value</i>	Allocates a specific amount of private memory for the <i>n</i> Galaxy partition.
lp_shared_mem_size	Allocates the shared memory for drivers in a Galaxy environment.
memory_test	Specifies the extent to which memory will be tested.
ocp_text	Overrides the default OCP display text with user-specified text.
os_type	Specifies the operating system and sets the appropriate console interface.
password	Sets a console password. Required for placing the SRM into secure mode.
pci_parity	Disables or enables parity checking on the PCI bus.
pk*0_fast	Enables fast SCSI mode.
pk*0_host_id	Specifies the default value for a controller host bus node ID.
pk*0_soft_term	Enables or disables SCSI terminators on systems that use the QLogic ISP1020 SCSI controller.
tt_allow_login	Enables or disables login to the SRM console firmware on other console ports.

2.25.1 auto_action

The **auto_action** environment variable specifies the action the console takes any time the system powers up, fails, or resets. The value of **auto_action** takes effect only after you reset the system by pressing the Reset button or by issuing the **init** command.

The default setting for **auto_action** is **halt**. With this setting, the system stops in the SRM console after being initialized. To cause the operating system to boot automatically after initialization, set the **auto_action** environment variable to **boot** or **restart**.

- When **auto_action** is set to **boot**, the system boots from the default boot device specified by the value of the **bootdef_dev** environment variable.
- When **auto_action** is set to **restart**, the system boots from whatever device it booted from before the shutdown/reset or failure.

NOTE: *For setting up an OpenVMS Galaxy environment, **auto_action** must be set to **halt**.*

NOTE: *After you set the **auto_action** environment variable, it is recommended that you set the boot device and operating system flags as well, using the **set bootdef_dev** and **set boot_osflags** commands.*

The syntax is:

set auto_action *value*

The options for value are:

halt	The system remains in console mode after power-up or a system crash.
boot	The operating system boots automatically after the SRM init command is issued or the Reset button is pressed.
restart	The operating system boots automatically after the SRM init command is issued or the Reset button is pressed, and it also reboots after an operating system crash.

Examples

In the following example, the operator sets the **auto_action** environment variable to **restart**. The device specified with the **bootdef_dev** environment variable is **dka0**. When UNIX is shut down and rebooted, the system will reboot from **dka0**.

```
P00>>> show auto_action
auto_action          halt
P00>>> set auto_action restart
P00>>> init
.
.
.
P00>>> show auto_action
auto_action          restart
P00>>> show bootdef_dev
bootdef_dev          dka0
P00>>> boot
...
```

(Log into UNIX and shutdown/reboot)

```
#shutdown -r now
...
console will boot from dka0
```

In the following example, **auto_action** is set to **restart**, but UNIX is booted from a device other than the device set with **bootdef_dev**. When UNIX is shut down and rebooted, the system reboots from the specified device.

```
P00>>> boot dka100
.
.
.
```

(Log into UNIX and shutdown/reboot)

```
#shutdown -r now
...
console will boot from dka100
```

2.25.2 bootdef_dev

The **bootdef_dev** environment variable specifies one or more devices from which to boot the operating system. When more than one device is specified, the system searches in the order listed and boots from the first device with operating system software.

Enter the **show bootdef_dev** command to display the current default boot device. Enter the **show device** command for a list of all devices in the system.

The syntax is:

set bootdef_dev *boot_device*

boot_device The name of the device on which the system software has been loaded. To specify more than one device, separate the names with commas.

Example

In this example, two boot devices are specified. The system will try booting from dkb0 and if unsuccessful, will boot from dka0.

```
P00>>> set bootdef_dev dkb0, dka0
```

NOTE: *When you set the **bootdef_dev** environment variable, it is recommended that you set the operating system boot parameters as well, using the **set boot_osflags** command.*

2.25.3 boot_file

The `boot_file` environment variable specifies the default file name to be used for booting when no file name is specified by the `boot` command. The factory default value is null.

The syntax is:

set boot_file *filename*

For systems running Linux, the *filename* is specific to the distribution of Linux:

- 2/boot/vmlinux.gz (Red Hat)
- 2/boot/vmlinuz (SuSE)

Example

In this example, a boot file is specified for booting the Redhat version of Linux.

```
P00>>> set boot_file 2/boot/vmlinux.gz
P00>>> boot
```

2.25.4 boot_osflags

The `boot_osflags` environment variable sets the default boot flags and, for OpenVMS, a root number.

Boot flags contain information used by the operating system to determine some aspects of a system bootstrap. Under normal circumstances, you can use the default boot flag settings.

To change the boot flags for the current boot only, use the *flags_value* argument with the **boot** command.

The syntax is:

set boot_osflags *flags_value*

The *flags_value* argument is specific to the operating system.

Tru64 UNIX Systems

Tru64 UNIX systems take a single ASCII character as the *flags_value* argument.

- a** Load operating system software from the specified boot device (autoboot). Boot to multiuser mode.
- i** Prompt for the name of a file to load and other options (boot interactively). Boot to single-user mode.
- s** Stop in single-user mode. Boots /vmunix to single-user mode and stops at the # (root) prompt.
- D** Full dump; implies “s” as well. By default, if UNIX crashes, it completes a partial memory dump. Specifying “D” forces a full dump at system crash.

Example

The following setting will autoboot Tru64 UNIX to multiuser mode when you enter the **boot** command.

```
P00>>> set boot_osflags a
```

Linux Systems

The *flags_value* argument for Linux on an ES40 system is:

```
"root=/dev/sda2"
```

Flags_value Arguments for Red Hat Distribution

- 0 Halt. (Do not set init default to this value.)
- 1 Single-user mode.
- 2 Multiuser, without NFS (same as 3, if you do not have networking)
- 3 Full multiuser mode (Default)
- 4 Unused
- 5 X11
- 6 Reboot. (Do not set init default to this value.)

Flags_value Arguments for SuSE

- 0 Halt. (Do not set init default to this value.)
- S Single-user mode. (Default)
- 1 Multi-user without network
- 2 Multiuser with network
- 3 Multiuser with network and xdm
- 6 Reboot. (Do not set init default to this value.)

Examples

Single-user mode is typically used for troubleshooting. To make system changes at this run level, you must have read/write privileges.

The following setting will boot Linux into single-user mode with read/write privileges under Red Hat distribution.

```
P00>>> set boot os_flags "root=/dev/sda2 1 rw"
```

The following setting will boot Linux into multiuser mode with network under SuSE distribution:

```
P00>>> set boot os_flags "root=/dev/sda2 2"
```

OpenVMS Systems

OpenVMS systems require an ordered pair as the *flags_value* argument: *root_number* and *boot_flags*.

root_number Directory number of the system disk on which OpenVMS files are located. For example:

<i>root_number</i>	Root Directory
--------------------	----------------

0 (default)	[SYS0.SYSEXEXE]
-------------	-----------------

1	[SYS1.SYSEXEXE]
---	-----------------

2	[SYS2.SYSEXEXE]
---	-----------------

3	[SYS3.SYSEXEXE]
---	-----------------

boot_flags The hexadecimal value of the bit number or numbers set. To specify multiple boot flags, add the flag values (logical OR). For example, the flag value 10080 executes both the 80 and 10000 flag settings. See Table 2–9.

Table 2–9 OpenVMS Boot Flag Settings

Flags_Value	Bit Number	Meaning
1	0	Bootstrap conversationally (enables you to modify SYSGEN parameters in SYSBOOT).
2	1	Map XDELTA to a running system.
4	2	Stop at initial system breakpoint.
8	3	Perform diagnostic bootstrap.
10	4	Stop at the bootstrap breakpoints.
20	5	Omit header from secondary bootstrap image.
80	7	Prompt for the name of the secondary bootstrap file.
100	8	Halt before secondary bootstrap.
10000	16	Display debug messages during booting.
20000	17	Display user messages during booting.

Examples

In the following OpenVMS example, *root_number* is set to 2 and *boot_flags* is set to 1. With this setting, the system will boot from root directory SYS2.SYSEXE to the SYSBOOT prompt when you enter the **boot** command.

```
P00>>> set boot_osflags 2,1
```

In the following OpenVMS example, *root_number* is set to 0 and *boot_flags* is set to 80. With this setting, you are prompted for the name of the secondary bootstrap file when you enter the **boot** command.

```
P00>>> set boot_osflags 0,80
```

2.25.5 com*_baud

The default baud rate for the system is 9600. The com*_baud commands set the baud rate for COM1 and COM2.

com1_baud

The **com1_baud** environment variable sets the baud rate for the internal COM1 serial interface.

com2_baud

The **com2_baud** environment variable sets the baud rate to match that of the device connected to the COM2 port.

The syntax is:

set com*_baud *baud_value*

baud_value The new baud rate. A list of possible values is displayed by entering the command without a value.

Example

The following example shows the supported baud rate values.

```
P00>>> set com2_baud
57600
38400
19200
9600
7200
4800
3600
2400
2000
1800
```

2.25.6 com*_flow

The com1_flow and com2_flow environment variables set the flow control on the COM1 and COM2 serial ports, respectively.

The syntax is:

set com*_flow *flow_value*

flow_value Defined values are:

none—No data flows in or out of the serial ports. Use this setting for devices that do not recognize XON/XOFF or that would be confused by these signals.

software—Use XON/XOFF(default). This is the setting for a standard serial terminal.

hardware—Use modem signals CTS/RTS. Use this setting if you are connecting a modem to a serial port.

Example

```
P00>>> set com1_flow hardware
```

2.25.7 com1_mode

The set com1_mode command specifies the COM1 data flow paths, so that data either flows through the RMC or bypasses it. You can also set com1_mode from the RMC.

By default all data passes through the RMC. Data and control signals flow from the system COM1 port, through the RMC, and to the active external port, either the COM1 serial port (MMJ) or the 9-pin modem port. If a modem is connected, the data goes to the modem. This mode is called through mode.

You can enter the RMC from either the MMJ port or the modem port. Only one session can be active at a time.

For modem connection, you can use the **set com1_mode** command to allow data to partially or completely bypass the RMC. The bypass modes are snoop mode, soft bypass mode, and firm bypass mode. These modes disable the local channel from sending characters to the system COM1 port.

You can also set the RMC to local mode, in which only the local channel can communicate with the system COM1 port. Local mode disables the modem from sending characters to the system COM1 port, but you can still enter the RMC.

The syntax is **set com1_mode** *value*

value Defined values are:

through All data passes through RMC and is filtered for the RMC escape sequence. This is the default.

snoop Data partially bypasses RMC, but RMC taps into the data lines and listens passively for the RMC escape sequence.

soft_bypass Data bypasses RMC, but RMC switches automatically into snoop mode if loss of carrier occurs.

firm_bypass Data bypasses RMC. RMC remote management features are disabled.

local Changes the focus of the COM1 traffic to the local RMC COM1 port if RMC is currently in one of the bypass modes or if RMC is in through mode with an active remote session.

Example

```
P00>>> set com1_mode snoop
```

2.25.8 com*_modem

The **com1_modem** and **com2_modem** environment variables are used to tell the operating system whether a modem is present on the COM1 or COM2 ports, respectively. From the settings of these variables, the operating system determines whether the port should assert a signal DTR.

The syntax is:

set com*_modem *modem_value*

modem_value Defined values are:

on—Modem is present.

off—Modem is not present (default value).

If you attach a modem to the COM1 or COM2 port, set the *modem_value* to **on**. This setting tells the operating system to assert a DTR (data terminal ready) signal to let the modem know that there is hardware attached. The modem responds with a DSR (data set ready) signal.

Example

```
P00>>> set com1_modem on
```

2.25.9 console

The console terminal can be either a VGA monitor or a serial terminal. The console environment variable specifies which type of console is used.

The syntax is:

set console *output_device*

The options for *output_device* are:

graphics (default)	The console terminal is a VGA monitor or a device connected to the VGA port.
serial	The console terminal is the device connected to the COM1 port.

The value of **console** takes effect only after you reset the system by pressing the Reset button or by issuing the **init** command.

Example

```
P00>>> show console
console                graphics
P00>>> set console serial
P00>>> init
.
.
.
P00>>> show console
console                serial
P00>>>
```

2.25.10 console_memory_allocation

The `console_memory_allocation` environment variable determines which memory locations the console will allocate for its private use.

Setting this environment variable to **new** allows the operating system to be loaded in the largest area of memory. Regardless of the setting, the console allocates the same amount of memory to the operating system. If **console_memory_allocation** is set to **old** with greater than 1gigabyte, memory is not contiguous (which may affect performance).

The value of **console_memory_allocation** takes effect only after you reset the system by pressing the Reset button or by issuing the **init** command.

Tru64 UNIX

ES40 systems running Tru64 UNIX V4.0F may encounter a Granularity Hint Regions (GH chunks) restriction. The following error message on the console terminal indicates possible performance loss caused by this restriction. Number (#) is a number that varies depending on memory size:

```
gh_chunks value of # invalid
```

To remove this restriction, follow the instructions in the *Tru64 UNIX Release Notes* for V4.0F.

OpenVMS

Set the **console_memory_allocation** to **new** before creating an OpenVMS Galaxy environment. See the DS20E-ES40 Remedial Kit available from <http://www.service.digital.com/patches> for instructions on creating a Galaxy on the ES40.

```
P00>>> set console_memory_allocation new
P00>>> init
.
.
.
```


The syntax is:

set console_memory_allocation *value*

The options for *value* are:

- old** For 1gigabyte or less, the console carves memory from 0–2 megabytes and at the end of memory, leaving all memory in between available to the operating system. If there is more than 1gigabyte, the console creates a “memory hole” for the operating system just under 1gigabyte.
- new** The console takes all needed memory from 0 megabytes to whatever amount is needed. It does not matter how much memory is installed and no holes are ever created.

2.25.11 `cpu_enabled`

The `cpu_enabled` environment variable sets a bit mask that enables or disables specific CPUs on a multiprocessor system.

Disabling a CPU may be necessary if a number of errors are reported on a specific CPU. These errors might be displayed during power-up or might be displayed with the **`show fru`** or **`show config`** command.

Disabled CPUs are prevented from running the console or the operating system. Bit 0 of the mask corresponds to CPU 0, bit 1 to CPU 1, and so on. A zero in the bit mask prevents the corresponding CPU from running; a one allows it to run. The bit mask is expressed as a hexadecimal value.

The value of **`cpu_enabled`** takes effect only after you reset the system by pressing the Reset button or by issuing the **`init`** command.

The **`cpu_enabled`** environment variable is typically used in benchmark testing.

NOTE: *The primary CPU cannot be disabled. The primary CPU is the lowest numbered working CPU.*

The syntax is:

set cpu_enabled *hex_digit*

The *hex_digit* values are shown in the table.

Hex_Digit Value	Binary Equivalent CPU enable 3210 (bit)	Enabled CPUs
0	0000	No CPUs (CPU 0 still comes up)
1	0001	CPU 0
2	0010	CPU 1
3	0011	CPU 0,1
4	0100	CPU 2
5	0101	CPU 0,2
6	0110	CPU 1,2
7	0111	CPU 0,1,2
8	1000	CPU 3
9	1001	CPU 0,3
A	1010	CPU 1,3
B	1011	CPU 0,1,3
C	1100	CPU 2,3
D	1101	CPU 0,2,3
E	1110	CPU 1,2,3
F	1111	CPU 0,1,2,3

Example

In the following example, CPU 0 and CPU 1 are enabled, and CPU 2 and CPU 3 are disabled.

```
P00>>> set cpu_enabled 3
```

2.25.12 ei*0_inet_init or ew*0_inet_init

The **ei*0_inet_init** or **ew*0_inet_init** environment variable determines whether the interface's internal Internet database is initialized from nvram or from a network server (through the bootp protocol). Legal values are nvram and bootp. The default value is bootp. Set this environment variable if you are booting UNIX from a RIS server.

To list the network devices on your system, enter the **show device** command. The Ethernet controllers start with the letters "ei" or "ew," for example, ewa0. The third letter is the adapter ID for the specific Ethernet controller. Replace the asterisk (*) with the adapter ID letter when using this command.

The syntax is:

set ei*0_inet_init *value* or
set ew*0_inet_init *value*

The *value* is one of the following:

nvram	Initializes the internal Internet database from nvram.
bootp	Initializes the internal Internet database from a network server through the bootp protocol.

Example

```
P00>>> set ewa0_inet_init bootp
```

2.25.13 ei*0_mode or ew*0_mode

The ei*0_mode or ew*0_mode environment variable sets an Ethernet controller to run an AUI, ThinWire, or twisted-pair Ethernet network. For the fast setting, the device defaults to fast.

To list the network devices on your system, enter the **show device** command. The Ethernet controllers start with the letters “ei” or “ew,” for example, ewa0. The third letter is the adapter ID for the specific Ethernet controller. Replace the asterisk (*) with the adapter ID letter when entering the command.

The syntax is:

set ei*0_mode *value* or
set ew*0_mode *value*

The options for *value* are:

au	Device type is AUI.
bnc	Device type is ThinWire.
fast	Device type is fast 100BaseT.
Fastfd	Device type is fast full duplex 100BaseT.
full	Device type is full duplex twisted-pair.
twisted-pair	Device type is 10BaseT (twisted-pair).

Example

```
P00>>> set ewa0_mode t
P00>>> show ewa0_mode
ewa0_mode          twisted-pair
```

2.25.14 ei*0_protocols or ew*0_protocols

The ei*0_protocols or ew*0_protocols environment variable sets network protocols for booting and other functions.

To list the network devices on your system, enter the **show device** command. The Ethernet controllers start with the letters “ei” or “ew,” for example, ewa0. The third letter is the adapter ID for the specific Ethernet controller. Replace the asterisk (*) with the adapter ID letter when entering the command.

The syntax is:

set ei*0_protocols *protocol_value* or
set ew*0_protocols *protocol_value*

The options for *protocol_value* are:

- | | |
|----------------------|---|
| mop (default) | Sets the network protocol to mop (Maintenance Operations Protocol), the setting typically used with the OpenVMS operating system. |
| bootp | Sets the network protocol to bootp, the setting typically used with the Tru64 UNIX operating system. |
| bootp,mop | When both are listed, the system attempts to use the mop protocol first, regardless of which is listed first. If not successful, it then attempts the bootp protocol. |

Example

```
P00>>> show device
.
.
.
ewa0.0.0.1001.0      EWA0      08-00-2B-3E-BC-B5
ewb0.0.0.12.0       EWB0      00-00-C0-33-E0-0D
ewc0.0.0.13.0       EWC0      08-00-2B-E6-4B-F3
.
.
.
P00>>> set ewa0_protocols bootp
P00>>> show ewa0_protocols
ewa0_protocols      bootp
```

2.25.15 kbd_hardware_type

The **kbd_hardware_type** environment variable sets the keyboard hardware type as either PCXAL or LK411 and enables the system to interpret the terminal keyboard layout correctly.

The syntax is:

set kbd_hardware_type *keyboard_type*

The options for *keyboard_type* are:

pcxal (default) Selects the 102-type keyboard layout.

lk411 Selects the LK411 keyboard layout.

Example

```
P00>>> set kbd_hardware_type lk411
P00>>>
```

2.25.16 language

The language environment variable specifies the keyboard layout, which depends on the language. The setting of the language environment variable must match the language of the keyboard variant.

The factory keyboard setting is 36 English (American).

The value of **language** takes effect only after you reset the system by pressing the Reset button or issuing the **init** command.

The syntax is:

set language *language_code*

The options for *language_code* are:

0	No language	42	Italiano (Italian)
30	Dansk (Danish)	44	Nederlands (Netherlands)
32	Deutsch (German)	46	Norsk (Norwegian)
34	Deutsch (Swiss)	48	Portugues (Portuguese)
36	English (American)	4A	Suomi (Finnish)
38	English (British/Irish)	4C	Svenska (Swedish)
3A	Español (Spanish)	4E	Belgisch-Nederlands (Dutch)
3C	Français (French)	50	Japanese (JIS)
3E	Français (Canadian)	52	Japanese (ANSI)
40	Français (Suisse Romande)		

Example

```
P00>>> set language 3A
```


2.25.17 lp_count

The `lp_count` environment variable specifies the number of partitions to be created in an OpenVMS Galaxy environment.

This environment variable is used in creating an OpenVMS Galaxy environment. See the DS20E-ES40 Remedial Kit available from <http://www.service.digital.com/patches> for information on creating a Galaxy on the ES40 system.

The syntax is:

set `lp_count` *value*

The *value* of **`lp_count`** represents the number of Galaxy partitions that will be created. The values are 0 (zero), 1, or 2.

- 0** If set to zero (0), Galaxy mode will be “off.” The system will boot a traditional SMP configuration.
- 1** If set to 1, one Galaxy partition will be created, and Galaxy features will be used. All CPUs will remain in partition 0.
- 2** If set to 2, two Galaxy partitions will be created, and Galaxy features will be used on both partitions.

Example

```
P00>>> set lp_count 2
```

2.25.18 lp_cpu_mask*n*

The `lp_cpu_maskn` environment variable specifies the bitmask of CPU IDs to be included in the *n* partition.

This environment variable is used in creating an OpenVMS Galaxy environment. See the DS20E-ES40 Remedial Kit available from <http://www.service.digital.com/patches> for information on creating a Galaxy on the ES40 system.

The syntax is:

set lp_cpu_mask*n* *value*

The *n* (number) is the specified partition, either zero (0) or 1. The *value* determines the CPUs to be assigned to that partition. The following restrictions apply:

- **lp_cpu_mask0** must have bit zero set (contains CPU 0).
- **lp_cpu_mask1** must have bit one set (contains CPU 1).

The table in Section 2.25.11 (**cpu_enabled**) lists CPU bitmask settings.

Example

This example of a four-CPU system assigns secondary CPU 2 with primary CPU 0 and secondary CPU 3 with primary CPU 1.

```
P00>>> set lp_cpu_mask0 5
P00>>> set lp_cpu_mask1 A
```

2.25.19 lp_io_maskn

The `lp_io_maskn` environment variable specifies the bitmask of I/O IDs to be included in the *n* partition.

This environment variable is used in creating an OpenVMS Galaxy environment. See the DS20E-ES40 Remedial Kit available from <http://www.service.digital.com/patches> for information on creating a Galaxy on the ES40 system.

The syntax is:

set lp_io_maskn *value*

The *n* (number) is the specified partition, either zero (0) or 1. The *value* determines the I/O IDs to be assigned to that partition. To assign I/O IDs in a two-instance Galaxy, **mask0** must be set to 1 and **mask1** must be set to 2.

Example

```
P00>>> set lp_io_mask0 1
P00>>> set lp_io_mask1 2
```

2.25.20 lp_mem_size*n*

The `lp_mem_sizen` environment variable allocates a specific amount of private memory for the *n* partition.

This environment variable is used in creating an OpenVMS Galaxy environment. See the DS20E-ES40 Remedial Kit available from <http://www.service.digital.com/patches> for information on creating a Galaxy on the ES40 system.

The syntax is:

set lp_mem_size*n* *value*

The *n* (number) is the specified partition, either zero (0) or 1. The *value* depends on how you decide to assign memory in your Galaxy configuration. The memory size values are specified in hexadecimal.

Example

In this example, 1000000 corresponds to 256 megabytes; C000000 corresponds to 192 megabytes.

```
P00>>> set lp_mem_size0 10000000
P00>>> set lp_mem_size1 C000000
```

2.25.21 lp_shared_mem_size

The `lp_shared_mem_size` environment variable allocates memory for use as shared memory.

This environment variable is used in creating an OpenVMS Galaxy environment. See the DS20E-ES40 Remedial Kit available from <http://www.service.digital.com/patches> for information on creating a Galaxy on the ES40 system.

Shared memory must be assigned in multiples of 8 megabytes. The minimum shared memory size is 8 megabytes. The memory size values are specified in hexadecimal. The syntax is:

set `lp_shared_mem_size` *value*

Example

In this example, 4000000 corresponds to 64 megabytes.

```
P00>>> set lp_shared_mem_size 4000000
```

2.25.22 memory_test

The `memory_test` environment variable determines the extent of memory testing on the next reset. You can set this variable for systems running UNIX.

The syntax is:

set memory_test *value*

The options for *value* are:

- | | |
|-----------------------|--|
| full (default) | Specifies that the full memory test will be run. Systems using the OpenVMS operating system must run the full memory test. |
| partial | Specifies that the first 256 MB of memory will be tested. |
| none | Specifies that memory will not be tested. (However, tests are always run on the first 32 MB.) |

Example

With the following setting, the first 256 MB of memory are tested when a system running Tru64 UNIX is reset.

```
P00>>> set memory_test partial
```

2.25.23 ocp_text

The `ocp_text` environment variable specifies a message to be displayed on the control panel display after self-tests and diagnostics have been completed. It is useful to set this environment variable if you have a number of systems and you want to identify each system by a node name.

The syntax is:

set ocp_text *message*

The *message* is the message you want to be displayed, typically the network node name you have defined for the system. The message can be up to 16 characters and must be entered in quotation marks.

Example:

```
P00>>> set ocp_text "Node Alpha1"  
P00>>>
```

2.25.24 os_type

The **os_type** environment variable specifies the default operating system. This variable is set at the factory to the setting for the operating system you purchased. Use this command to change the factory default setting.

The value of **os_type** takes effect only after you reset the system by pressing the Reset button or by issuing the **init** command.

The syntax is:

set os_type *os_type*

The options for *os_type* are:

- | | |
|--------------|--|
| unix | Sets the default to UNIX. The SRM firmware is started during power-up or reset. |
| vms | Sets the default to OpenVMS. The SRM firmware is started during power-up or reset. |
| linux | Sets the default to Linux. The SRM firmware is started during power-up or reset. |

Example

In this example, the default operating system is set to Tru64 UNIX. After the system is initialized, the UNIX banner is displayed.

```
P00>>> set os_type unix
P00>>> init
.
.
.
```


2.25.25 pci_parity

The `pci_parity` environment variable disables or enables parity checking on the PCI bus.

Some PCI devices do not implement PCI parity checking, and some have a parity-generating scheme in which the parity is sometimes incorrect or is not fully compliant with the PCI specification. A side effect of this behavior is that superfluous PCI parity errors are reported by the host PCI bridge. In such cases, the device can be used as long as parity is not checked.

CAUTION: *Disabling PCI parity checking on this system is not recommended or supported.*

The syntax is:

set pci_parity *value*

The options for *value* are:

on (default) Enables PCI parity checking.

off Disables PCI parity checking.

Example

```
P00>>> show pci_parity
pci parity          on
```

2.25.26 pk*0_fast

The `pk*0_fast` environment variable enables fast SCSI to perform in either standard or fast mode.

If the system has at least one fast SCSI device, set the default controller speed to fast SCSI (1). Devices on a controller that connects to both standard and fast SCSI devices will perform at the appropriate rate for the device. If the system has no fast SCSI devices, set the default controller speed to standard SCSI (0). If a fast SCSI device is on a controller set to standard, it will perform in standard mode.

To list the controllers on your system, enter the **show device** command. SCSI controllers begin with the letters “pk,” for example, pka0. The third letter is the adapter ID for the specific SCSI controller. Replace the asterisk with the adapter ID letter when entering the **set pk*0_fast** command.

The value of **set pk*0_fast** takes effect only after you reset the system by pressing the Reset button or by issuing the **init** command.

The syntax is:

set pk*0_fast *scsi_speed*

The options for *scsi_speed* are:

0 The controller is in standard SCSI mode.

1 (default) The controller is in fast SCSI mode.

Example

```
P00>>> set pkb0_fast 1
P00>>> init
.
.
.
P00>>> show pkb0_fast
P00>>> pkb0_fast                    1
```

2.25.27 pk*0_host_id

The `pk*0_host_id` environment variable sets the controller host bus node ID to a value between 0 and 7.

Each SCSI bus in the system requires a controller. Buses can support up to eight devices; however, the eighth device must be a controller. Each device on the bus, including the controller, must have a unique ID, which is a number between 0 and 7. This is the bus node ID number.

On each bus, the default bus node ID for the controller is set to 7. You do not need to change the controller bus node ID unless you place two or more controllers on the same bus.

To list the controllers on your system, enter the **show device** command. SCSI controllers begin with the letters “pk” (for example, pka0). The third letter is the adapter ID for the controller. Replace the asterisk with the adapter ID letter when entering the **set pk*0_host_id** command.

The value of **pk*0_host_id** takes effect only after you reset the system by pressing the Reset button or by issuing the **init** command.

The syntax is:

set pk*_host_id *scsi_node_id*

The value for *scsi_node_id* is the bus node ID, a number from 0 to 7.

Example

In this example, the default bus node ID for a SCSI controller with an adapter ID of “b” is set to bus node ID 6.

```
P00>>> set pkb0_host_id 6
P00>>> init
.
.
.
P00>>> show pkb0_host_id
pkb0_host_id          6
```

2.25.28 pk*0_soft_term

The `pk*0_soft_term` environment variable enables or disables SCSI terminators for optional SCSI controllers. This environment variable applies to systems that use the QLogic SCSI controller, though it does not affect the onboard controller.

The QLogic ISP1020 SCSI controller implements the 16-bit wide SCSI bus. The QLogic module has two terminators, one for the low eight bits and one for the high eight bits.

To list the controllers on your system, enter the **show device** command. SCSI controllers begin with the letters “pk” (for example, pka0). The third letter is the adapter ID for the controller. Replace the asterisk with the adapter ID letter when entering the **set pk*0_soft_term** command.

The value of **pk*0_soft_term** takes effect only after you reset the system by pressing the Reset button or by issuing the **init** command.

The syntax is:

set pk*0_soft_term *value*

The options for *value* are:

- | | |
|---------------------|--|
| off | Disables termination of all 16 bits. |
| low | Enables low eight bits and disables high eight bits. |
| high | Enables high eight bits and disables low eight bits. |
| on (default) | Enables all 16 bits. |

Examples

In this example, both terminators are disabled.

```
P00>>> set pkb0_soft_term off
P00>>> init
.
.
.
P00>>> show pkb0_soft_term
pkb0_soft_term          off
```

In this example, the terminator for the high 8 bits is enabled.

```
P00>>> set pkb0_soft_term high
P00>>> init
.
.
.
P00>>> show pkb0_soft_term
pkb0_soft_term          high
```

2.25.29 tt_allow_login

The **tt_allow_login** environment variable enables or disables login to the SRM console firmware on alternative console ports. “Login” refers to pressing the Return or Enter key to activate the console device.

If the **console** environment variable is set to **serial**, the primary console device is the terminal connected through the COM1 port. The **set tt_allow_login 1** command lets you activate a console device through COM2 or a VGA monitor. The **set tt_allow_login 0** command disables console activation through alternative ports. You might want to disable console access to COM2 as a system security measure or if you want to use COM2 as an “application only” port.

The syntax is:

set tt_allow_login *value*

The options for *value* are:

- 0** Disables login through the COM2 port or the VGA monitor.
- 1** (default) Enables login through the COM2 port or the VGA monitor.

Example

In the following example, the primary console device is set to the terminal connected through the COM1 port. Then the **set tt_allow_login 0** command is used to disable logins through either the COM2 port or a VGA monitor.

```
P00>>> set console serial
P00>>> init
.
.
.
P00>> set tt_allow_login 0
```

2.26 Ensuring Console Security

The SRM console firmware has console security features intended to prevent unauthorized personnel from modifying the system parameters or otherwise tampering with the system from the console. The security features include a secure mode and commands to set console security.

2.26.1 Overview of Secure Mode

The SRM console has two modes, user mode and secure mode.

- User mode allows you to use all SRM console commands. User mode is the default mode.
- Secure mode allows you to use only the **boot** and **continue** commands. The **boot** command cannot take command-line parameters when the console is in secure mode. The console boots the operating system using the environment variables stored in NVRAM (**boot_file**, **bootdef_dev**, **boot_flags**).

Secure Function Commands

- The **set password** and **set secure** commands are used to set secure mode.
 - The **clear password** command is used to exit secure mode and return to user mode. All the SRM console commands are available and the console is no longer secure.
 - The **login** command turns off console security for the current console session. Once you enter the **login** command in secure mode, you can enter any SRM command as usual. However, the system automatically returns to secure mode when you enter the **boot** or **continue** command or when you reset the system.
-

NOTE: *The security features work only if access to the system hardware is denied to unauthorized personnel. Be sure the system is available only to authorized personnel.*

2.26.2 Setting the Console Password

Set the console password with the set password command. A password is required for operating the system in secure mode.

Example 2-23 Set Password

```
P00>>> set password ❶
Please enter the password:
Please enter the password again:
P00>>>

P00>>> set password ❷
Please enter the password:
Please enter the password again:
Now enter the old password:
P00>>>

P00>>> set password
Please enter the password:
Password length must be between 15 and 30 characters ❸
P00>>>
```


- ❶ Setting a password. If a password has not been set and the **set password** command is issued, the console prompts for a password and verification. The password and verification are not echoed.
- ❷ Changing a password. If a password has been set and the **set password** command is issued, the console prompts for the new password and verification, then prompts for the old password. The password is not changed if the validation password entered does not match the existing password stored in NVRAM.
- ❸ The password length must be between 15 and 30 alphanumeric characters. Any characters entered after the 30th character are not stored.

The **set password** command sets the console password for the first time or changes an existing password. It is necessary to set the password only if the system is going to operate in secure mode.

The syntax is:

set password

2.26.3 Setting the Console to Secure Mode

To set the console to secure mode, first set the password. Then enter the set secure command. The system immediately enters secure mode.

Example 2-24 Set Secure

```
P00>>> set secure ❶  
Console is secure. Please login.  
P00>>> b dkb0  
Console is secure - parameters are not allowed.  
P00>>> login ❷  
Please enter the password:  
P00>>> b dkb0  
(boot dkb0.0.0.3.1)  
.  
.  
.
```

- ❶ The console is put into secure mode, and then the operator attempts to boot the operating system with command-line parameters. A message is displayed indicating that boot parameters are not allowed when the system is in secure mode.
- ❷ The **login** command is entered to turn off security features for the current console session. After successfully logging in, the operator enters a **boot** command with command-line parameters.

The **set secure** command enables secure mode. If no password has been set, you are prompted to set the password. Once you set a password and enter the **set secure** command, secure mode is in effect immediately and only the **continue**, **boot** (using the stored parameters), and **login** commands can be performed.

The syntax is:

set secure

2.26.4 Turning Off Security During a Console Session

The `login` command turns off the security features, enabling access to all SRM console commands during the current console session. The system automatically returns to secure mode as soon as the `boot` or `continue` command is entered or when the system is reset.

Example 2-25 Login

```
P00>>> login                                ❶
Secure not set. Please set the password.
P00>>> set password                          ❷
Please enter the password:
Please enter the password again:
P00>>> login                                ❸
Please enter the password.
P00>>> show boot*
```

- ❶ The **`login`** command is entered, but the system is not in secure mode. A password must be set.
- ❷ A password is set.
- ❸ The **`login`** command is entered. After the password is entered, console security is turned off for the current session and the operator can enter commands.

When you enter the **login** command, you are prompted for the current system password. If a password has not been set, a message is displayed indicating that there is no password in NVRAM. If a password has been set, this prompt is displayed:

Please enter the password:

If the password entered matches the password in NVRAM, when the prompt is redisplayed the console is no longer in secure mode and all console commands can be performed during the current console session.

If You Forget the Password

If you forget the current password, use the **login** command in conjunction with the control panel Halt button to clear the password, as follows:

1. Enter the **login** command:

```
P00>>> login
```

2. When prompted for the password, press the Halt button to the latched position and then press the Return (or Enter) key.
3. Press the Halt button to release the halt. The password is now cleared and the console cannot be put into secure mode unless you set a new password.

2.26.5 Returning to User Mode

The `clear password` command clears the password environment variable, setting it to zero. Once the password is cleared, you are returned to user mode.

Example 2-26 Clear Password

```
P00>>> clear password
Please enter the password: ❶
Console is secure
P00>>> clear password
Please enter the password: ❷
Password successfully cleared.
P00>>>
```

- ❶ The wrong password is entered. The system remains in secure mode.
- ❷ The password is successfully cleared.

The **`clear password`** command is used to exit secure mode and return to user mode. To use **`clear password`**, you must know the current password. Once you clear the password, the console is no longer secure.

To clear the password without knowing the current password, you must use the **`login`** command in conjunction with the Halt button, as described in Section 2.26.4.

2.27 Updating Firmware

System firmware is typically updated whenever the operating system is updated. You might also need to update firmware if you add I/O device controllers and adapters; if enhancements are made to the firmware; or if the serial ROM or RMC firmware should ever become corrupted.

Procedures for updating the SRM firmware are described in the *ES40 Owner's Guide*.

Chapter 3

AlphaBIOS Console

AlphaBIOS is the enhanced BIOS graphical interface for running certain utilities on AlphaServer systems.

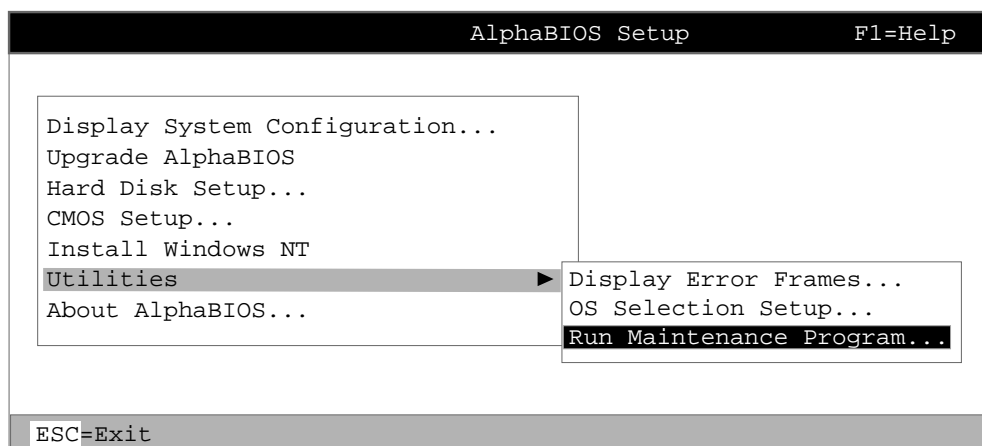
This chapter explains how to invoke AlphaBIOS to run utility programs. Sections in this chapter are:

- Starting AlphaBIOS
- Running Utility Programs

3.1 Starting AlphaBIOS

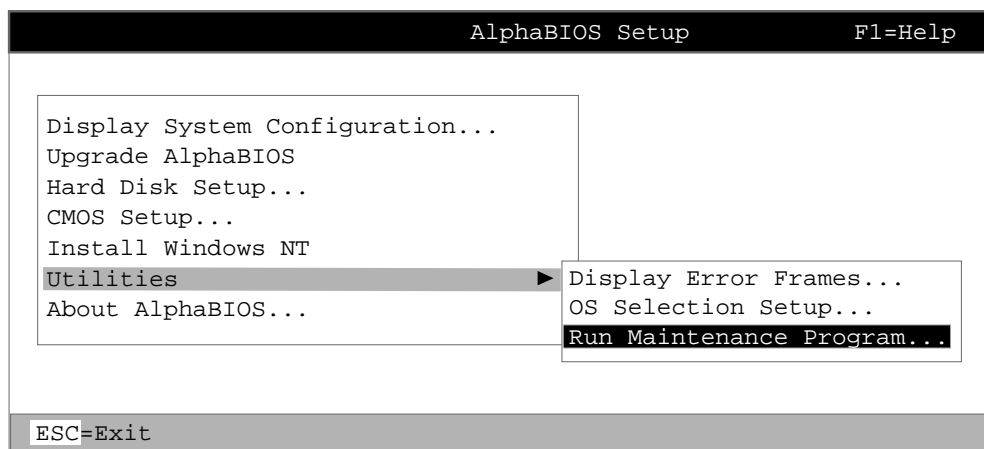
To start AlphaBIOS, power up the system to the SRM prompt and enter the `alphabios` command. A startup screen displays information about the system and processors. Press F2 to enter the Setup screen. From the Setup screen, select Utilities. See Figure 3-1 and Figure 3-2.

Figure 3-1 AlphaBIOS Startup Screen



PK0954a

Figure 3-2 Utilities Menu

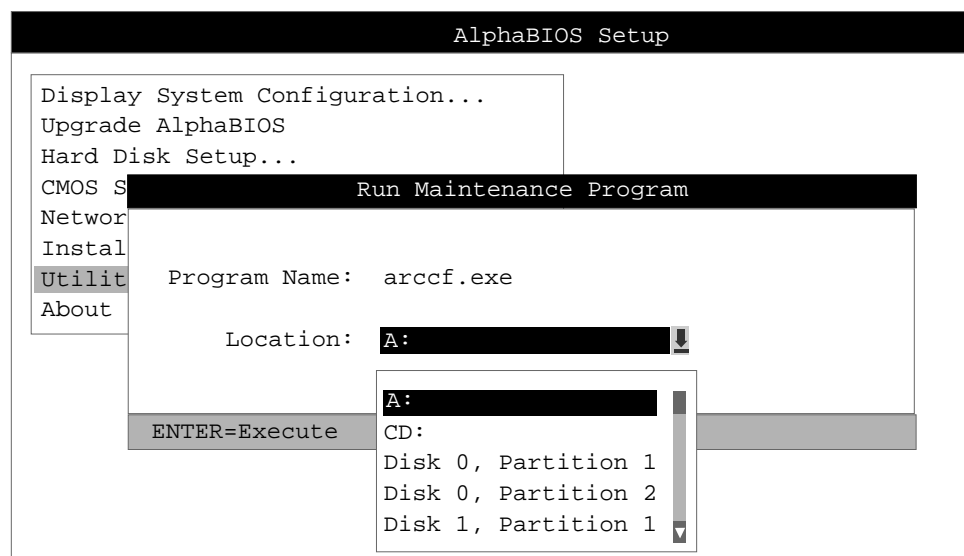


PK0954a

3.2 Running a Maintenance Program

From the Utilities menu, enter the name of the program to be run. Enter the location if you do not specify the path with the program name. Figure 3-3 shows the program name entry for the RAID configuration utility.

Figure 3-3 Run Maintenance Program Dialog Box



PK0929

Maintenance programs can be run from either a VGA monitor or a serial terminal. If you have a serial terminal, see Chapter 2 of the *ES40 Owner's Guide* for directions.

If your system has a RAID controller installed, and you change your system configuration (for example, by adding another RAID drive), you will have to run a RAID configuration utility. As you modify your system, you might be required to run other types of maintenance programs as well.

As shown in Figure 3-3 the Program Name is the name of the program to be run. It must be an executable (.EXE). When entering the program name, you do not need to type the extension. Programs run from AlphaBIOS must be written as ARC compatible images.

The Location is the location from which the program will be run if no path is entered along with the program name. To display a list of all available disks and partitions, press the Alt and down arrow keys simultaneously. Once the list is displayed, use the arrow keys to cycle through the selections.

Chapter 4

RMC Commands

This chapter describes the command set for the remote management console (RMC). The operation of the RMC is described in the *ES40 Owner's Guide*.

4.1 RMC Commands List

From the remote management console command-line interface, you can issue commands to monitor the system (voltages, temperatures, and fans), and manipulate it (reset, power on/off, halt) remotely or locally.

clear {alert, port}
deposit
disable {alert, remote}
dump
enable {alert, remote}
env
halt {in, out}
hangup
help or ?
power {on, off}
quit
reset
send alert
set {alert, com1_mode, dial, escape, init, logout, password, user}
status

NOTE: *The **dep** and **dump** commands are reserved.*

Command Conventions

Observe the following conventions for entering RMC commands:

- Enter enough characters to distinguish the command.

NOTE: *The **reset** and **quit** commands are exceptions. You must enter the entire word for these commands to work.*

- For commands consisting of two words, enter the entire first word and at least one letter of the second word. For example, you can enter **disable a** for **disable alert**.
- For commands that have parameters, you are prompted for the parameter.
- Use the Backspace key to erase input.
- If you enter a nonexistent command or a command that does not follow conventions, the following message is displayed:

```
*** ERROR - unknown command ***
```

- If you enter a string that exceeds 14 characters, the following message is displayed:

```
*** ERROR - overflow ***
```

4.2 clear alert

The clear alert command clears the current alert condition and causes the RMC to stop paging the remote system operator.

If the alert is not cleared, the RMC continues to page the remote operator every 30 minutes if the dial-out alert feature is enabled.

The **clear alert** command clears the current alert so that the RMC can capture a new alert. The last alert is stored until a new event overwrites it. The Alert Pending field of the **status** command becomes NO after the alert is cleared.

```
RMC> clear alert  
RMC>
```

4.3 clear port

The clear port command clears any “stuck” conditions on the system’s COM1 port.

The **clear port** command attempts to free the port by resetting all UARTs controlled by the RMC if the port is currently locked by an application program, without resetting the entire system.

```
RMC> clear port  
RMC>
```

NOTE: *This command also causes the modem to disconnect.*

4.4 disable alert

The disable alert command disables the RMC from paging a remote system operator.

Monitoring continues and alerts are still logged in the Last Alert field of the **status** command, but alerts are not sent to the remote operator.

```
RMC> disable alert  
RMC>
```

4.5 disable remote

The disable remote command disables remote access to the RMC modem port and disables automatic dial-out.

```
RMC> disable remote  
RMC>
```

4.6 enable alert

The `enable alert` command enables the RMC to page a remote system operator.

Before you can enter the **enable alert** command, you must configure remote dial-in and call-out, set an RMC password, and enable remote access to the RMC modem port. See the *ES40 Owner's Guide* for the complete set-up procedure.

```
RMC> set dial
Dial String: ATXDT9,15085553333
RMC> set alert
Alert String: ,,,,,,5085553332#;
RMC> enable alert
RMC>
```

If the **enable alert** command fails, the following error message is displayed:

```
*** ERROR - enable failed ***
```

Issue the **status** command to see if the Remote Access field is set to Enabled.

4.7 enable remote

The enable remote command enables remote access to the RMC modem port by configuring the modem with the setting stored in the initialization string.

This command also allows the RMC to automatically dial the pager number set with the **set dial** command upon detection of alert conditions.

Before you can enter the **enable remote** command, you must configure remote dial-in by setting an RMC password and initialization string. See the *ES40 Owner's Guide* for the complete set-up procedure.

```
RMC> set password
RMC Password: ****
Verification: ****
RMC> set init
Init String: AT&F0E0V0X0S0=2
RMC> enable remote
```

If the **enable remote** command fails, the following error message is displayed:

```
*** ERROR - enable failed ***
```

Check that the modem is connected and that you have set the initialization string correctly.

4.8 env

The env command displays the system environmental status, including power supplies, voltages, fans, and temperatures. If a fault has occurred, the reading blinks.

RMC>env

```
System Hardware Monitor

Temperature (warnings at 45.0°C, power-off at 50.0°C)
  CPU0: 26.0°C   CPU1: 26.0°C   CPU2: 27.0°C   CPU3: 26.0°C
  Zone0: 29.0°C   Zone1: 30.0°C   Zone2: 31.0°C
Fan RPM
  Fan1: 2295   Fan2: 2295   Fan3: 2205
  Fan4: 2235   Fan5: OFF    Fan6: 2518
Power Supply(OK, FAIL, OFF, '----' means not present)
  PS0 : OK     PS1 : OK     PS2 : ----
  CPU0: OK     CPU1: OK     CPU2: OK     CPU3: OK
CPU CORE voltage
  CPU0: +2.192V   CPU1: +2.192V   CPU2: +2.192V   CPU3: +2.192V
CPU IO voltage
  CPU0: +1.488V   CPU1: +1.488V   CPU2: +1.488V   CPU3: +1.488V
Bulk voltage
  +3.3V Bulk: +3.328V   +5V Bulk: +5.076V   +12V Bulk: +12.096V
  Vterm: +1.824V       Cterm: +2.000V     -12V Bulk: -12.480V
```

❶

❷

❸

❹

❺

❻

- ❶ CPU temperature. In this example four CPUs are present.
- ❷ Temperature of PCI backplane: Zone 0 includes PCI slots 1–3, Zone 1 includes PCI slots 7–10, and Zone 2 includes PCI slots 4–6.
- ❸ Fan RPM. With the exception of Fan 5, all fans are powered as long as the system is powered on. Fan 5 is OFF unless Fan 6 fails.
- ❹ The normal power supply status is either OK (system is powered on) or OFF (system is powered off or the power supply cord is not plugged in). FAIL indicates a problem with a supply.
- ❺ CPU CORE voltage and CPU I/O voltage. In a good system, the core voltage for all CPUs should be the same, and the I/O voltage for all CPUs should be the same.
- ❻ Bulk power supply voltage.

4.9 halt in

The halt in command is equivalent to pressing the Halt button on the control panel.

The **halt in** command halts the managed system. When the **halt in** command is issued, the terminal exits RMC and returns to the server's COM1 port.

Toggling the Power button on the operator control panel overrides the **halt in** condition.

```
RMC> halt in
Returning to COM port
```

4.10 halt out

The halt out command is equivalent to releasing the Halt button on the control panel.

The **halt out** command releases a halt. The terminal exits RMC and returns to the server's COM1 port.

```
RMC> halt out
Returning to COM port
```

You cannot use **halt out** to release a halt if the Halt button on the operator control panel is latched in. If you issue the command, the following message is displayed:

```
RMC> halt out
Halt button is IN
```

4.11 hangup

The hangup command terminates the modem session.

If you do not issue the **hangup** command, the session is disconnected automatically after a period of idle time set by the **set logout** command. The default is 20 minutes.

```
RMC> hangup  
RMC>
```

4.12 help or ?

The help or ? command displays the RMC command set.

```
RMC> help
clear {alert, port}
deposit
disable {alert, remote}
dump
enable {alert, remote}
env
halt {in, out}
hangup
help or ?
power {off, on}
quit
reset
send alert
set {alert, coml_mode, dial, escape, init, logout, password, user}
status
```

4.13 power off

The power off command is equivalent to turning off the system power from the operator control panel.

If the system is already powered off, this command has no effect. You can override the **power off** command either by issuing a **power on** command or by toggling the Power button on the operator control panel.

```
RMC> power off  
RMC>
```

4.14 power on

The power on command is equivalent to turning on the system power from the operator control panel.

If the system is already powered on, this command has no effect. After the **power on** command is issued, the terminal exits RMC and reconnects to the server's COM1 port.

```
RMC> power on
Returning to COM port
```

The **power on** command does not turn on the system if the Power button on the operator control panel is in the OFF position. If you issue the command, the following message is displayed:

```
RMC> power on
Power button is OFF
```

4.15 quit

The quit command exits RMC and returns the terminal to the server's COM1 port.

You must enter the entire word for the command to take effect.

```
RMC> quit  
Returning to COM port
```


4.16 reset

The reset command is equivalent to pushing the Reset button on the operator control panel.

The **reset** command restarts the system. The terminal exits RMC and reconnects to the server's COM1 port. You must enter the entire word for the command to take effect.

```
RMC>reset
Returning to COM port
```

4.17 send alert

The send alert command forces an alert condition.

This command is used to test the setup of the dial-out alert function. It is issued from the local terminal.

As long as no one connects to the modem and there is no alert pending, the alert will be sent to the pager immediately.

If the pager does not receive the alert, re-check your setup.

```
RMC> send alert
Alert detected!
```

4.18 set alert

The set alert command sets the alert string that is transmitted through the modem when an alert condition is detected.

Set the alert string to the phone number of the modem connected to the remote system. The alert string is appended after the dial string, and the combined string is sent to the modem.

The alert string consists of the following elements:

- ,,,,, Each comma (,) provides a 2-second delay. In this example, a delay of 12 seconds is set to allow the paging service to answer.
- 5085553332# A call-back number for the paging service. The alert string must be terminated by the pound (#) character.
- ; A semicolon (;) must be used to terminate the entire string.

The example shown below is generic. Because paging services vary, be sure to listen to the options provided by the paging service to determine the appropriate delay and the menu options.

```
RMC> set alert
Alert String: ,,,,,,5085553332#;
RMC>
```

4.19 set com1_mode

The set com1_mode command specifies the COM1 data flow paths, so that data either passes through the RMC or bypasses it.

By default all data passes through the RMC. Data and control signals flow from the system COM1 port, through the RMC, and to the active external port, either the COM1 serial port (MMJ) or the 9-pin modem port. If a modem is connected, the data goes to the modem. This mode is called Through mode.

You can enter the RMC from either the MMJ port or the modem port. Only one session can be active at a time.

For modem connection, you can set the **com1_mode** environment variable to allow data to partially or completely bypass the RMC. The bypass modes are Snoop mode, Soft Bypass mode, and Firm Bypass mode. These modes disable the local channel from sending characters to the system COM1 port. If the **com1_mode** value has been set to **soft_bypass** or **firm_bypass**, and the system is turned off, the mode reverts to Snoop.

- In Snoop mode, you can type an escape sequence to enter the RMC. RMC mode provides a command-line interface for issuing commands to monitor and control the system.
- In Soft Bypass mode, you cannot enter the RMC. But if an alert condition or loss of carrier occurs, the RMC switches into Snoop mode. From Snoop mode you can enter RMC.
- In Firm Bypass mode you cannot enter the RMC. To enter, reset the **com1_mode** environment variable from the SRM console, as described in Chapter 2, then set up the RMC again from the local terminal.

You can also set the RMC to Local mode, in which only the local channel can communicate with the system COM1 port. Local mode disables the modem from sending characters to the system COM1 port, but you can still get into the RMC.

NOTE: *You can always enter the RMC locally regardless of the current mode.*

You can set **com1_mode** to one of the following values:

through	All data passes through RMC and is filtered for the escape sequence. This is the default.
snoop	Data partially bypasses RMC, but RMC taps into the data lines and listens passively for the escape sequence.
soft_bypass	Data bypasses RMC, but RMC switches automatically into Snoop mode if loss of carrier occurs.
firm_bypass	Data bypasses RMC. RMC remote management features are disabled.
local	Changes the focus of the COM1 traffic to the local MMJ port if RMC is currently in one of the bypass modes or is in Through mode with an active remote session.

Example

```
RMC> set com1_mode  
Com1_mode (THROUGH, SNOOP, SOFT_BYPASS, FIRM_BYPASS, LOCAL): local
```

4.20 set dial

The set dial command sets the string to be used by the RMC to dial out when an alert condition occurs.

The dial string must be in the correct format for the attached modem. If a paging service is to be contacted, the string must include the appropriate modem commands to dial the number. The dial string is case sensitive. The RMC automatically converts all alphabetic characters to uppercase.

The dial string consists of the following elements:

ATXDT	AT = Attention. X = Forces the modem to dial “blindly” (not seek the dial tone). Enter this character if the dial-out line modifies its dial tone when used for services such as voice mail. D = Dial T = Tone (for touch-tone)
9,	The number for an outside line (in this example, 9). Enter the number for an outside line if your system requires it. , = Pause for 2 seconds.
15085553333	Phone number of the paging service.

```
RMC> set dial
Dial String: ATXDT9,15085553333
RMC>
```

4.21 set escape

The set escape command sets a new escape sequence for invoking RMC.

The escape sequence can be any character string, not to exceed 14 characters. A typical sequence consists of two or more control characters. It is recommended that control characters be used in preference to ASCII characters. Use the **status** command to verify the escape sequence.

Be sure to record the new escape sequence. If you forget the escape sequence, you must reset the RMC to the factory defaults. See the *ES40 Owner's Guide* for information on setting the RMC to the factory defaults.

The following example consists of two instances of the Esc key and the letters "FUN." The "F" is not displayed when you set the sequence because it is preceded by the escape character. Enter the **status** command to see the new escape sequence.

```
RMC> set escape
Escape Sequence: un
RMC> status
.
.
.
Escape Sequence: ^[^FUN
```

4.22 set init

The set init command sets the modem initialization string.

The initialization string is limited to 31 characters and can be modified, depending on the type of modem used.

```
RMC> set init
Init String: AT&F0E0V0X0S0=2
RMC>
```

Because the modem commands disallow mixed cases, the RMC automatically converts all alphabetic characters entered in the init string to uppercase.

The RMC automatically configures the modem's flow control according to the setting of the SRM **com1_flow** environment variable. The RMC also enables the modem carrier detect feature to monitor the modem connectivity.

4.23 set logout

The set logout command sets the amount of time before the RMC terminates an inactive modem connection. The default is 20 minutes.

The settings are in tens of minutes, 0–9. The zero (0) setting disables logout. With logout disabled, the RMC never disconnects the idle modem session.

The following example sets the logout timer to 30 minutes.

```
RMC> set logout
Logout Time (0-9 tens of minutes): 3
```

4.24 set password

The set password command allows you to set or change the password that is prompted for at the beginning of a modem session.

A password must be set to enable access through a modem. The string cannot exceed 14 characters. For security, the password is not echoed on the screen. When prompted for verification, type the password again. If you mistype, reenter the **set password** command.

```
RMC> set pass
RMC Password: ****
Verification: ****
*** ERROR - Verification failed, password is not set ***
RMC> set pass
RMC Password: ****
Verification: ****
```

4.25 set user

The set user command allows you to set a user string to be displayed in the status command.

You may want to make notes regarding the system. The string is limited to 63 characters and is displayed in the User String field when you enter the **status** command.

In this example, the operator leaves a reminder that a power supply needs to be replaced.

```
RMC> set user
User String: need to replace P/S
RMC> status
  PLATFORM STATUS
.
.
.
User String: need to replace P/S
```

4.26 status

The status command displays the system status and the current RMC settings.

Table 4–1 describes each field of the **status** command output.

```
RMC> status
PLATFORM STATUS
On-Chip Firmware Revision: V1.0
Flash Firmware Revision: V2.2
Server Power: ON
System Halt: Deasserted
RMC Power Control: ON
Escape Sequence: ^^[RMC
Remote Access: Enabled
RMC Password: set
Alert Enable: Disabled
Alert Pending: YES
Init String: AT&F0E0V0X0S0=2
Dial String: ATXDT9,15085553333
Alert String: ,,,,,,5085553332#;
Com1_mode: THROUGH
Last Alert: PS1 failed
Logout Timer: 20 minutes
User String: need to replace PS
```

Table 4–1 Status Command Fields

Field	Meaning
On-Chip Firmware Revision:	Revision of RMC firmware on the microcontroller.
Flash Firmware Revision:	Revision of RMC firmware in flash ROM.
Server Power:	ON = System is on. OFF = System is off.
System Halt:	Asserted = System has been halted. Deasserted = Halt has been released.
RMC Power Control:	ON= System has powered on from RMC. OFF = System has powered off from RMC.
Escape Sequence:	Current escape sequence for access to RMC console.
Remote Access:	Enabled = Modem for remote access is enabled. Disabled = Modem for remote access is disabled.
RMC Password:	Set = Password set for modem access. Not set = No password set for modem access.
Alert Enable:	Enabled = Dial-out enabled for sending alerts. Disabled = Dial-out disabled for sending alerts.
Alert Pending:	YES = Alert has been triggered. NO = No alert has been triggered.
Init String:	Initialization string that was set for modem.
Dial String:	Pager string to be dialed when an alert occurs.
Alert String:	Identifies the system that triggered the alert to the paging service. Usually the phone number of the monitored system.
Com1_mode:	Identifies the current COM1 mode.
Last Alert:	Type of alert (for example, power supply 1 failed).
Logout Timer:	The amount of time before the RMC terminates an inactive modem connection. The default is 20 minutes.
User String:	Notes supplied by user.

Index

A

- alphabios command (SRM), 2-39
- AlphaBIOS console, 3-1
 - defined, 1-4
 - loading, 2-39
 - maintenance programs, 3-4
 - starting, 3-2
- AlphaBIOS Utilities menu, 3-3
- auto_action environment variable, 2-2, 2-49
- auto_action environment variable, SRM, 2-37
- Autoboot, 2-49

B

- Baud rate, setting, 2-57
- boot command (SRM), 2-26
- Boot devices, specifying, 2-51
- Boot file, specifying, 2-52
- Boot flags
 - OpenVMS, 2-55
 - UNIX, 2-53
- boot_file environment variable, 2-52
- boot_osflags environment variable, 2-53
- Bootable devices, displaying, 2-17
- bootdef_dev environment variable, 2-51
- bootp protocol, 2-69
- Bus node ID, SCSI, 2-82

C

- clear alert command (RMC), 4-4
- clear password command (SRM), 2-93
- clear port command (RMC), 4-5
- Clearing SRM password, 2-93
- COM ports, baud rate, 2-57
- com*_baud environment variable, 2-57
- com*_flow environment variable, 2-58

- com*_modem environment variable, 2-61
- com1_baud environment variable, 2-57
- com1_flow environment variable, 2-58
- com1_mode environment variable, 2-59
- com1_modem environment variable, 2-61
- com2_baud environment variable, 2-57
- com2_flow environment variable, 2-58
- com2_modem environment variable, 2-61
- Command conventions, RMC, 4-3
- Command syntax, SRM console, 2-6
- Console commands list (SRM), 2-4
- Console device, activating, 2-85
- console environment variable, 2-62
- Console mode, 1-4
- Console password, 2-87
 - clearing, 2-93
- Console program, 1-2
- Console terminal, 1-3
- Console tests, 2-31
- Console, specifying, 2-62
- console_memory_allocation
 - environment variable, 2-63
- Control panel message, 2-78
- Controllers, SCSI, 2-82
- CPU, enabling, 2-65
- cpu_enabled environment variable, 2-65
- crash command (SRM), 2-32
- Crash dump, 2-32

D

- Device naming, 2-17
- Devices, verifying, 2-30
- Diagnostic tests, 2-30
- disable alert command (RMC), 4-6
- disable remote command (RMC), 4-7

Displaying logical configuration, 2-12
DTR, asserting, 2-61

E

edit command (SRM), 2-41
ei*0_inet_init environment variable,
2-67
ei*0_mode environment variable, 2-68
ei*0_protocols environment variable,
2-69
enable alert command (RMC), 4-8
enable remote command (RMC), 4-9
env command (RMC), 4-10
Environment variables
initializing, 2-45
summary, 2-46
Escape sequence, RMC, 2-43
Ethernet controllers, 2-68
Ethernet settings, 2-68
Event log, displaying, 2-35
ew*0_inet_init environment variable, 2-
67
ew*0_mode environment variable, 2-68
ew*0_protocols environment variable,
2-69

F

Fans, status of, 2-24
Fast SCSI, 2-81
File, displaying, 2-35
Firmware
updating, 2-94
FRUs, displaying, 2-18
FRUs, displaying errors, 2-21

G

Galaxy
lp_count environment variable, 2-72
lp_cpu_mask*n* environment variable,
2-73
lp_io_mask*n* environment variable,
2-74
lp_mem_size*n* environment variable,
2-75

lp_shared_mem_size environment
variable, 2-76
galaxy command. *See* lpinit command

H

Halt button, with login command, 2-92
halt in command (RMC), 4-12
halt out command (RMC), 4-13
hangup command (RMC), 4-14
Hardware, initializing, 2-36
help command (SRM), 2-10
help or ? command (RMC), 4-15

I

init command (SRM), 2-36
Initializing hardware in a Galaxy, 2-38
Initializing the system, 2-36
Internet database, initializing, 2-67

K

kbd_hardware_type environ. variable,
2-70
Keyboard
language variants, 2-71
Keyboard type, setting, 2-70

L

language environment variable, 2-71
login command (SRM), 2-91
Loopback tests, 2-31
lp_count environment variable, 2-72
lp_cpu_mask environment variable, 2-
73
lp_io_mask environment variable, 2-74
lp_mem_size environment variable, 2-
75
lp_shared_mem_size environment
variable, 2-76
lpinit command (SRM), 2-38

M

Maintenance programs, 3-4
man command (SRM), 2-10

- Memory test
 - SRM, 2-77
- Memory, displaying, 2-22
- memory_test environment variable, 2-77
- MOP protocol, 2-69
- MOP V3 software, 2-41
- more command (SRM), 2-35

N

- Nvram script, 2-40

O

- ocp_text environment variable, 2-78
- Operating system, specifying, 2-79
- os_type environment variable, 2-79

P

- PALcode version, displaying, 2-23
- Parity checking, 2-80
- Password, setting SRM, 2-87
- PCI NVRAM module, configuring, 2-28
- PCI options, slot numbers, 2-16
- PCI parity, 2-80
- pci_parity environment variable, 2-80
- pk*0_fast environment variable, 2-81
- pk*0_host_id environment variable, 2-82
- pk*0_soft_term environment variable, 2-83
- power off command (RMC), 4-16
- power on command (RMC), 4-17
- Power status, displaying, 2-24
- Power-up memory test
 - SRM, 2-77
- Power-up script, creating, 2-40
- prcache command (SRM), 2-28
- Processor, enabling, 2-65
- Program, resuming, 2-34

Q

- QLogic controller, 2-83
- quit command (RMC), 4-18

R

- RAID configuration, 3-5
- RAID devices, configuring, 2-29
- Reading a file, 2-35
- reset command (RMC), 2-43, 4-19
- Resetting firmware, 2-36
- RMC
 - command conventions, 4-3
 - defined, 1-5
 - escape sequence, 2-43
- rmc command (SRM), 2-42
- RMC commands
 - clear alert, 4-4
 - clear port, 4-5
 - disable alert, 4-6
 - disable remote, 4-7
 - enable alert, 4-8
 - enable remote, 4-9
 - env, 4-10
 - halt in, 4-12
 - halt out, 4-13
 - hangup, 4-14
 - help or ?, 4-15
 - list of, 4-2
 - power off, 4-16
 - power on, 4-17
 - quit, 4-18
 - reset, 2-43, 4-19
 - send alert, 4-20
 - set alert, 4-21
 - set com1_mode, 4-22
 - set dial, 4-24
 - set escape, 4-25
 - set init, 4-26
 - set logout, 4-27
 - set password, 4-28
 - set user, 4-29
 - status, 4-30
- Running utilities in AlphaBIOS, 3-3

S

- SCSI controller IDs, 2-82
- SCSI devices, speed of, 2-81
- SCSI terminators, enabling, 2-83
- Secure function commands, 2-86

- Secure mode, setting SRM to, 2-89
- Security features, turning off, 2-91
- Security, SRM console, 2-86
- send alert command (RMC), 4-20
- Serial ports
 - determining presence of modem on, 2-61
 - flow control, 2-58
- set alert command (RMC), 4-21
- set com1_mode command (RMC), 4-22
- set command (SRM), 2-44, 2-45
- set dial command (RMC), 4-24
- set escape command (RMC), 4-25
- set init command (RMC), 4-26
- set logout command (RMC), 4-27
- set password command (RMC), 4-28
- set password command (SRM), 2-87
- set secure command (SRM), 2-89
- set user command (RMC), 4-29
- show command (SRM), 2-44, 2-45
- show config command (SRM), 2-12
- show device command (SRM), 2-17
- show error command (SRM), 2-21
- show fru command (SRM), 2-18
- show memory command (SRM), 2-22
- show pal command (SRM), 2-23
- show power command (SRM), 2-24
- show version command (SRM), 2-25
- Special characters, SRM console, 2-7
- SRM console
 - command syntax, 2-6
 - defined, 1-4
 - device naming conventions, 2-17
 - displaying system configuration, 2-12
 - finding help, 2-10
 - invoking, 2-2
 - invoking, 2-2
 - invoking from AlphaBIOS, 2-2
 - invoking from RMC, 2-3
 - loading AlphaBIOS, 2-39
 - reading a file, 2-35
 - resetting firmware, 2-36
 - security, 2-86
 - setting environment variables, 2-44
 - special characters, 2-7
 - tasks performed from, 2-9

- testing the system, 2-30
- version, 2-25
- SRM console commands, 2-4
 - alphabios, 2-39
 - boot, 2-26
 - clear password, 2-93
 - crash, 2-32
 - edit, 2-41
 - help, 2-10
 - init, 2-36
 - login, 2-91
 - lpinit, 2-38
 - man, 2-10
 - more, 2-35
 - prcache, 2-28
 - rmc, 2-42
 - set, 2-44, 2-45
 - set password, 2-87
 - set secure, 2-89
 - show, 2-44, 2-45
 - show config, 2-12
 - show device, 2-17
 - show error, 2-21
 - show fru, 2-18
 - show memory, 2-22
 - show pal, 2-23
 - show power, 2-24
 - show version, 2-25
 - test, 2-30
- SRM console password, 2-87
 - clearing, 2-93
- Starting AlphaBIOS, 3-2
- status command (RMC), 4-30

T

- Temperature sensors, status of, 2-24
- test command (SRM), 2-30
- Testing memory from SRM, 2-77
- Testing the system, 2-30
- tt_allow_login environment variable, 2-85

U

- Updating firmware, 2-94
- Utilities menu, AlphaBIOS, 3-3

V

VGA console test, 2-31

VGA monitor, invoking RMC from,
2-42

